

Transactions on Computational Science
and Computational Intelligence

Hamid R. Arabnia · Leonidas Deligiannidis
Michael R. Grimaila · Douglas D. Hodson
Kazuki Joe · Masakazu Sekijima
Fernando G. Tinetti *Editors*

Advances in Parallel & Distributed Processing, and Applications

Proceedings from PDPTA'20, CSC'20,
MSV'20, and GCC'20

 Springer

Transactions on Computational Science and Computational Intelligence

Series Editor

Hamid R. Arabnia

Department of Computer Science

The University of Georgia

Athens, GA, USA

Computational Science (CS) and Computational Intelligence (CI) both share the same objective: finding solutions to difficult problems. However, the methods to the solutions are different. The main objective of this book series, “Transactions on Computational Science and Computational Intelligence”, is to facilitate increased opportunities for cross-fertilization across CS and CI. This book series will publish monographs, professional books, contributed volumes, and textbooks in Computational Science and Computational Intelligence. Book proposals are solicited for consideration in all topics in CS and CI including, but not limited to, Pattern recognition applications; Machine vision; Brain-machine interface; Embodied robotics; Biometrics; Computational biology; Bioinformatics; Image and signal processing; Information mining and forecasting; Sensor networks; Information processing; Internet and multimedia; DNA computing; Machine learning applications; Multi-agent systems applications; Telecommunications; Transportation systems; Intrusion detection and fault diagnosis; Game technologies; Material sciences; Space, weather, climate systems, and global changes; Computational ocean and earth sciences; Combustion system simulation; Computational chemistry and biochemistry; Computational physics; Medical applications; Transportation systems and simulations; Structural engineering; Computational electro-magnetic; Computer graphics and multimedia; Face recognition; Semiconductor technology, electronic circuits, and system design; Dynamic systems; Computational finance; Information mining and applications; Astrophysics; Biometric modeling; Geology and geophysics; Nuclear physics; Computational journalism; Geographical Information Systems (GIS) and remote sensing; Military and defense related applications; Ubiquitous computing; Virtual reality; Agent-based modeling; Computational psychometrics; Affective computing; Computational economics; Computational statistics; and Emerging applications.

For further information, please contact Mary James, Senior Editor, Springer, mary.james@springer.com.

More information about this series at <http://www.springer.com/series/11769>

Hamid R. Arabnia • Leonidas Deligiannidis
Michael R. Grimaila • Douglas D. Hodson
Kazuki Joe • Masakazu Sekijima
Fernando G. Tinetti
Editors

Advances in Parallel & Distributed Processing, and Applications

Proceedings from PDPTA'20, CSC'20,
MSV'20, and GCC'20

 Springer

Editors

Hamid R. Arabnia
Department of Computer Science
University of Georgia
Athens, GA, USA

Leonidas Deligiannidis
School of Computing and Data Sciences
Wentworth Institute of Technology
Boston, MA, USA

Michael R. Grimaila
Center for Cyberspace Research (CCR)
Air Force Institute of Technology
Wright-Patterson AFB, OH, USA

Douglas D. Hodson
Electrical and Computer Engineering
Air Force Institute of Technology
Wright-Patterson AFB, OH, USA

Kazuki Joe
Information and Computer Sciences
Nara Women's University
Nara, Japan

Masakazu Sekijima
School of Computing
Tokyo Institute of Technology
Meguro City, Tokyo, Japan

Fernando G. Tinetti
Facultad de Informática - CIC PBA
Universidad Nacional de La Plata
La Plata, Argentina

ISSN 2569-7072

ISSN 2569-7080 (electronic)

Transactions on Computational Science and Computational Intelligence

ISBN 978-3-030-69983-3

ISBN 978-3-030-69984-0 (eBook)

<https://doi.org/10.1007/978-3-030-69984-0>

© Springer Nature Switzerland AG 2021

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

It gives us great pleasure to introduce this collection of papers that were presented at the following international conferences: Scientific Computing (CSC 2020); Parallel & Distributed Processing Techniques and Applications (PDPTA 2020); Modeling, Simulation & Visualization Methods (MSV 2020); and Grid, Cloud, & Cluster Computing (GCC 2020). These four conferences were held simultaneously (same location and dates) at Luxor Hotel (MGM Resorts International), Las Vegas, USA, July 27–30, 2020. This international event was held using a hybrid approach, that is, “in-person” and “virtual/online” presentations and discussions.

This book is composed of ten Parts. Parts I through IV (composed of 27 chapters) include articles that address various challenges in the area of scientific computing (CSC). Parts V through VII (composed of 31 chapters) include articles that discuss advances in the area of parallel and distributed processing (PDPTA). Recent progress in the fields of modeling, simulation, and visualization methods (MSV) appear in Parts VIII through IX (composed of 17 chapters). Lastly, Part X (composed of 10 chapters) presents advances in grid, cloud, and cluster computing (GCC).

An important mission of the World Congress in Computer Science, Computer Engineering, and Applied Computing, CSCE (a federated congress to which this event is affiliated with), includes “*Providing a unique platform for a diverse community of constituents composed of scholars, researchers, developers, educators, and practitioners. The Congress makes concerted effort to reach out to participants affiliated with diverse entities (such as: universities, institutions, corporations, government agencies, and research centers/labs) from all over the world. The congress also attempts to connect participants from institutions that have **teaching** as their main mission with those who are affiliated with institutions that have **research** as their main mission. The congress uses a quota system to achieve its institution and geography diversity objectives.*” By any definition of diversity, this congress is among the most diverse scientific meeting in the USA. We are proud

to report that this federated congress had authors and participants from 54 different nations representing variety of personal and scientific experiences that arise from differences in culture and values.

The program committees (refer to subsequent pages for the list of the members of committees) would like to thank all those who submitted papers for consideration. About 50% of the submissions were from outside the USA. Each submitted paper was peer reviewed by two experts in the field for originality, significance, clarity, impact, and soundness. In cases of contradictory recommendations, a member of the conference program committee was charged to make the final decision; often, this involved seeking help from additional referees. In addition, papers whose authors included a member of the conference program committee were evaluated using the double-blind review process. One exception to the above evaluation process was for papers that were submitted directly to chairs/organizers of pre-approved sessions/workshops; in these cases, the chairs/organizers were responsible for the evaluation of such submissions. The overall paper acceptance rate for regular papers was 20%; 18% of the remaining papers were accepted as short and/or poster papers.

We are grateful to the many colleagues who offered their services in preparing this book. In particular, we would like to thank the members of the Program Committees of individual research tracks as well as the members of the Steering Committees of CSC 2020, PDPTA 2020, MSV 2020, and GCC 2020; their names appear in the subsequent pages. We would also like to extend our appreciation to over 500 referees.

As sponsors-at-large, partners, and/or organizers, each of the followings (separated by semicolons) provided help for at least one research track: Computer Science Research, Education, and Applications (CSREA); US Chapter of World Academy of Science; American Council on Science and Education & Federated Research council; and Colorado Engineering Inc. In addition, a number of university faculty members and their staff, several publishers of computer science and computer engineering books and journals, chapters and/or task forces of computer science associations/organizations from three regions, and developers of high-performance machines and systems provided significant help in organizing the event as well as providing some resources. We are grateful to them all.

We express our gratitude to all authors of the articles published in this book and the speakers who delivered their research results at the congress. We would also like to thank the followings: UCMSS (Universal Conference Management Systems & Support, California, USA) for managing all aspects of the conference; Dr. Tim Field of APC for coordinating and managing the printing of the programs; the staff of Luxor Hotel (MGM Convention) for the professional service they provided; and Ashu M. G. Solo for his help in publicizing the congress. Last but not least, we would like to thank Ms. Mary James (Springer Senior Editor in New York) and

Arun Pandian KJ (Springer Production Editor) for the excellent professional service they provided for this book project.

Athens, GA, USA

Hamid R. Arabnia

Boston, MA, USA

Leonidas Deligiannidis

Wright-Patterson AFB, OH, USA

Ryan D. Engle

Wright-Patterson AFB, OH, USA

Michael R. Grimaila

Wright-Patterson AFB, OH, USA

Douglas D. Hodson

Nara City, Japan

Kazuki Joe

Meguro City, Tokyo, Japan

Masakazu Sekijima

Chofu, Tokyo, Japan

Hayaru Shouno

La Plata, Argentina

Fernando G. Tinetti

Scientific Computing

CSC 2020 – Program Committee

- *Prof. Abbas M. Al-Bakry (Steering Committee); University of IT and Communications, Baghdad, Iraq*
- *Prof. Emeritus Nizar Al-Holou (Steering Committee); ECE Department; Vice Chair; IEEE/SEM-Computer Chapter; University of Detroit Mercy, Detroit, Michigan, USA*
- *Prof. Emeritus Hamid R. Arabnia (Steering Committee); University of Georgia, USA; Editor-in-Chief, Journal of Supercomputing (Springer); Fellow, Center of Excellence in Terrorism, Resilience, Intelligence & Organized Crime Research (CENTRIC).*
- *Dr. Azita Bahrami (Co-Editor, EEE); President, IT Consult, USA*
- *Prof. Dr. Juan-Vicente Capella-Hernandez; Universitat Politècnica de València (UPV), Department of Computer Engineering (DISCA), Valencia, Spain*
- *Prof. Emeritus Kevin Daimi (Steering Committee); Department of Mathematics, Computer Science and Software Engineering, University of Detroit Mercy, Detroit, Michigan, USA*
- *Prof. Zhangisina Gulnur Davletzhanovna; Central-Asian University, Kazakhstan, Almaty; Vice President of International Academy of Informatization, Kazskhstan, Almaty, Republic of Kazakhstan*
- *Prof. Leonidas Deligiannidis (Steering Committee); Department of Computer Information Systems, Wentworth Institute of Technology, Boston, Massachusetts, USA*
- *Dr. Ryan D. Engle; US Air Force Institute of Technology (AFIT), USA*
- *Prof. George A. Gravvanis (Steering Committee); Director, Physics Laboratory & Head of Advanced Scientific Computing, Applied Math & Applications Research Group; Professor of Applied Mathematics and Numerical Computing and Department of ECE, Democritus University of Thrace, Xanthi, Greece.*
- *Prof. Michael R. Grimaila (Steering Committee); US Air Force Institute of Technology (AFIT), USA*

- *Prof. Houcine Hassan; Department of Computer Engineering (Systems Data Processing and Computers), Universitat Politecnica de Valencia, Spain*
- *Dr. Douglas D. Hodson (Steering Committee); US Air Force Institute of Technology (AFIT), USA*
- *Prof. George Jandieri (Steering Committee); Georgian Technical University, Tbilisi, Georgia; Chief Scientist, The Institute of Cybernetics, Georgian Academy of Science, Georgia; Ed. Member, International Journal of Microwaves and Optical Technology, The Open Atmospheric Science Journal, American Journal of Remote Sensing, Georgia*
- *Prof. Dr. Abdeldjalil Khelassi; CS Department, Abou beker Belkaid University of Tlemcen, Algeria; Editor-in-Chief, Medical Tech. Journal; Assoc. Editor, Electronic Physician Journal (EPJ) - Pub Med Central*
- *Prof. Byung-Gyu Kim (Steering Committee); Multimedia Processing Communications Lab.(MPCL), Department of CSE, College of Engineering, SunMoon University, South Korea*
- *Prof. Louie Lolong Lacatan; Chairperson, Computer Engineering Department, College of Engineering, Adamson University, Manila, Philippines; Senior Member, International Association of Computer Science and Information Technology (IACSIT), Singapore; Member, IAOE, Austria*
- *Dr. Andrew Marsh (Steering Committee); CEO, HoIP Telecom Ltd (Healthcare over Internet Protocol), UK; Secretary General of World Academy of BioMedical Sciences and Technologies (WABT) a UNESCO NGO, The United Nations*
- *Dr. Ali Mostafaeipour; Industrial Engineering Department, Yazd University, Yazd, Iran*
- *Dr. Housseem Eddine Nouri; Informatics Applied in Management, Institut Supérieur de Gestion de Tunis, University of Tunis, Tunisia*
- *Prof. Dr., Eng. Robert Ehimen Okonigene (Steering Committee); Department of Electrical & Electronics Engineering, Faculty of Engineering and Tech., Ambrose Alli University, Edo State, Nigeria*
- *Prof. James J. (Jong Hyuk) Park (Steering Committee); Department of Computer Science and Engineering (DCSE), SeoulTech; President, FTRA, EiC, HCIS Springer, JoC, IJITCC; Head of DCSE, SeoulTech, Korea*
- *Dr. Akash Singh (Steering Committee); IBM Corporation, Sacramento, California, USA; Chartered Scientist, Science Council, UK; Fellow, British Computer Society; Member, Senior IEEE, AACR, AAAS, and AAI; IBM Corporation, USA*
- *Ashu M. G. Solo (Publicity), Fellow of British Computer Society, Principal/R&D Engineer, Maverick Technologies America Inc.*
- *Prof. Fernando G. Tinetti (Steering Committee); School of Computer Science, Universidad Nacional de La Plata, La Plata, Argentina; also at Comision Investigaciones Cientificas de la Prov. de Bs. As., Argentina*
- *Prof. Hahanov Vladimir (Congress Steering Committee); Vice Rector, and Dean of the Computer Engineering Faculty, Kharkov National University of Radio Electronics, Ukraine and Professor of Design Automation Department, Computer Engineering Faculty, Kharkov; IEEE Computer Society Golden Core Member; National University of Radio Electronics, Ukraine*

- *Prof. Shiuh-Jeng Wang (Steering Committee); Director of Information Cryptology and Construction Laboratory (ICCL) and Director of Chinese Cryptology and Information Security Association (CCISA); Department of Information Management, Central Police University, Taoyuan, Taiwan; Guest Ed., IEEE Journal on Selected Areas in Communications.*
- *Prof. Layne T. Watson (Steering Committee); Fellow of IEEE; Fellow of The National Institute of Aerospace; Professor of Computer Science, Mathematics, and Aerospace and Ocean Engineering, Virginia Polytechnic Institute & State University, Blacksburg, Virginia, USA*
- *Prof. Jane You (Steering Committee); Associate Head, Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong*
- *Prof. Dr., Eng. Robert Ehimen Okonigene (Steering Committee); Department of Electrical & Electronics Engineering, Faculty of Engineering and Technology, Ambrose Alli University, Nigeria*
- *Chiranjibi Sitaula; Head, Department of Computer Science and IT, Ambition College, Kathmandu, Nepal*
- *Dr. Yunlong Wang; Advanced Analytics at QuintilesIMS, Pennsylvania, USA*

Parallel & Distributed Processing Techniques and Applications

PDPTA 2020 – Program Committee

- *Prof. Emeritus Hamid R. Arabnia (Steering Committee); University of Georgia, USA; Editor-in-Chief, Journal of Supercomputing (Springer); Fellow, Center of Excellence in Terrorism, Resilience, Intelligence & Organized Crime Research (CENTRIC).*
- *Dr. P. Balasubramanian; School of CSE, Nanyang Technological University, Singapore*
- *Prof. Dr. Juan-Vicente Capella-Hernandez; Universitat Politecnica de Valencia (UPV), Department of Computer Engineering (DISCA), Valencia, Spain*
- *Prof. Juan Jose Martinez Castillo; Director, The Acanelys Alan Turing Nikola Tesla Research Group and GIPEB, Universidad Nacional Abierta, Venezuela*
- *Prof. Emeritus Kevin Daimi (Steering Committee); Department of Mathematics, Computer Science and Software Engineering, University of Detroit Mercy, Detroit, Michigan, USA*
- *Prof. Leonidas Deligiannidis (Steering Committee); Department of Computer Information Systems, Wentworth Institute of Technology, Boston, Massachusetts, USA*
- *Prof. Mary Mehrnoosh Eshaghian-Wilner (Steering Committee); Professor of Engineering Practice, University of Southern California, California, USA; Adjunct Professor, Electrical Engineering, University of California Los Angeles, Los Angeles (UCLA), California, USA*
- *Prof. Houcine Hassan; Department of Computer Engineering (Systems Data Processing and Computers), Universitat Politecnica de Valencia, Spain*
- *Prof. Hiroshi Ishii; Department Chair, Tokai University, Minato, Tokyo, Japan*
- *Prof. Makoto Iwata; School of Information, Kochi University of Technology, Kami, Kochi, Japan*
- *Prof. George Jandieri (Steering Committee); Georgian Technical University, Tbilisi, Georgia; Chief Scientist, The Institute of Cybernetics, Georgian Academy of Science, Georgia; Ed. Member, International Journal of Microwaves and*

Optical Technology, The Open Atmospheric Science Journal, American Journal of Remote Sensing, Georgia

- *Prof. Kazuki Joe (Steering Committee); Nara Women's University Nara, Japan*
- *Prof. Byung-Gyu Kim (Steering Committee); Multimedia Processing Communications Lab.(MPCL), Department of CSE, College of Engineering, SunMoon University, South Korea*
- *Prof. Tai-hoon Kim; School of Information and Computing Science, University of Tasmania, Australia*
- *Prof. Louie Lolong Lacatan; Chairperson, CE Department, College of Engineering, Adamson University, Manila, Philippines; Senior Member, International Association of Computer Science and Information Technology (IACSIT), Singapore; Member, International Association of Online Engineering (IAOE), Austria*
- *Prof. Dr. Guoming Lai; Computer Science and Technology, Sun Yat-Sen University, Guangzhou, P. R. China*
- *Prof. Hyo Jong Lee; Director, Center for Advanced Image and Information Technology, Division of Computer Science and Engineering, Chonbuk National University, South Korea*
- *Dr. Andrew Marsh (Steering Committee); CEO, HoIP Telecom Ltd (Healthcare over Internet Protocol), UK; Secretary General of World Academy of BioMedical Sciences and Technologies (WABT) a UNESCO NGO, The United Nations*
- *Prof. Salahuddin Mohammad Masum; Computer Engineering Technology, Southwest Tennessee Community College, Memphis, Tennessee, USA*
- *Dr. Ali Mostafaeipour; Industrial Engineering Department, Yazd University, Yazd, Iran*
- *Prof. Hiroaki Nishikawa; Faculty of Engineering, Information and Systems, University of Tsukuba, Japan*
- *Prof. Dr., Eng. Robert Ehimen Okonigene (Steering Committee); Department of Electrical & Electronics Engineering, Faculty of Engineering and Technology, Ambrose Alli University, Nigeria*
- *Prof. James J. (Jong Hyuk) Park (Steering Committee); DCSE, SeoulTech, Korea; President, FTRA, EiC, HCIS Springer, JoC, IJITCC; Head of DCSE, SeoulTech, Korea*
- *Dr. Prantosh K. Paul; Department of CIS, Raiganj University, Raiganj, West Bengal, India*
- *Prof. Dr. R. Ponalagusamy; Department of Mathematics, National Institute of Technology, India*
- *Dr. Masakazu Sekijima; Tokyo Institute of Technology, Japan*
- *Dr. Manik Sharma; Department of Computer Science and Applications, DAV University, Jalandhar, India*
- *Prof. Hayaru Shouno (Steering Committee); The University of Electro-Communications, Japan*
- *Ashu M. G. Solo (Publicity), Fellow of British Computer Society, Principal/R&D Engineer, Maverick Technologies America Inc.*

- *Prof. Fernando G. Tinetti (Steering Committee); School of CS, Universidad Nacional de La Plata, La Plata, Argentina; also at Comision Investigaciones Cientificas de la Prov. de Bs. As., Argentina*
- *Prof. Hahanov Vladimir (Steering Committee); Vice Rector, and Dean of the Computer Engineering Faculty, Kharkov National University of Radio Electronics, Ukraine and Professor of Design Automation Department, Computer Engineering Faculty, Kharkov; IEEE Computer Society Golden Core Member; National University of Radio Electronics, Ukraine*
- *Dr. Haoxiang Harry Wang; Cornell University, Ithaca, New York, USA; Founder and Director, GoPerception Laboratory, New York, USA*
- *Prof. Shiuh-Jeng Wang (Steering Committee); Director of Information Cryptology and Construction Laboratory (ICCL) and Director of Chinese Cryptology and Information Security Association (CCISA); Department of Information Management, Central Police University, Taoyuan, Taiwan; Guest Ed., IEEE Journal on Selected Areas in Communications.*
- *Prof. Layne T. Watson (Steering Committee); Fellow of IEEE; Fellow of The National Institute of Aerospace; Professor of Computer Science, Mathematics, and Aerospace and Ocean Engineering, Virginia Polytechnic Institute & State University, Blacksburg, Virginia, USA*
- *Prof. Jane You (Steering Committee); Associate Head, Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong*

Modeling, Simulation & Visualization Methods

MSV 2020 – Program Committee

- *Prof. Emeritus Nizar Al-Holou (Steering Committee); Electrical and Computer Engineering Department; Vice Chair, IEEE/SEM-Computer Chapter; University of Detroit Mercy, Detroit, Michigan, USA*
- *Prof. Hamid R. Arabnia (Steering Committee); University of Georgia, USA; Editor-in-Chief, Journal of Supercomputing (Springer); Fellow, Center of Excellence in Terrorism, Resilience, Intelligence & Organized Crime Research (CENTRIC).*
- *Prof. Emeritus Kevin Daimi (Steering Committee); Department of Mathematics, Computer Science and Software Engineering, University of Detroit Mercy, Detroit, Michigan, USA*
- *Prof. Leonidas Deligiannidis (Steering Committee); Department of Computer Information Systems, Wentworth Institute of Technology, Boston, Massachusetts, USA*
- *Prof. Byung-Gyu Kim (Steering Committee); Multimedia Processing Communications Lab.(MPCL), Department of CSE, College of Engineering, SunMoon University, South Korea*
- *Prof. Hyo Jong Lee; Director, Center for Advanced Image and Information Technology, Division of Computer Science and Engineering, Chonbuk National University, South Korea*
- *Dr. Muhammad Naufal Bin Mansor; Faculty of Engineering Technology, Department of Electrical, Universiti Malaysia Perlis (UniMAP), Perlis, Malaysia*
- *Prof. Aree Ali Mohammed; Head, Computer Science Department, University of Sulaimani, Kurdistan, Iraq*
- *Prof. James J. (Jong Hyuk) Park (Steering Committee); DCSE, SeoulTech, Korea; President, FTRA, EiC, HCIS Springer, JoC, IJITCC; Head of DCSE, SeoulTech, Korea*
- *Dr. Xuewei Qi; Research Faculty & PI, Center for Environmental Research and Technology, University of California, Riverside, California, USA*

- *Ashu M. G. Solo (Publicity), Fellow of British Computer Society, Principal/R&D Engineer, Maverick Technologies America Inc.*
- *Prof. Fernando G. Tinetti (Steering Committee); School of CS, Universidad Nacional de La Plata, La Plata, Argentina; also at Comision Investigaciones Cientificas de la Prov. de Bs. As., Argentina*
- *Dr. Haoxiang Harry Wang; Cornell University, Ithaca, New York, USA; Founder and Director, GoPerception Laboratory, New York, USA*
- *Prof. Shiuh-Jeng Wang (Steering Committee); Director of Information Cryptology and Construction Laboratory (ICCL) and Director of Chinese Cryptology and Information Security Association (CCISA); Department of Information Management, Central Police University, Taoyuan, Taiwan; Guest Ed., IEEE Journal on Selected Areas in Communications.*
- *Prof. Layne T. Watson (Steering Committee); Fellow of IEEE; Fellow of The National Institute of Aerospace; Professor of Computer Science, Mathematics, and Aerospace and Ocean Engineering, Virginia Polytechnic Institute & State University, Blacksburg, Virginia, USA*
- *Prof. Jane You (Steering Committee); Associate Head, Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong*

Grid, Cloud, & Cluster Computing

GCC 2020 – Program Committee

- *Prof. Emeritus Nizar Al-Holou (Steering Committee); ECE Department; Vice Chair; IEEE/SEM-Computer Chapter; University of Detroit Mercy, Detroit, Michigan, USA*
- *Prof. Hamid R. Arabnia (Steering Committee); University of Georgia, USA; Editor-in-Chief, Journal of Supercomputing (Springer); Editor-in-Chief, Transactions of Computational Science & Computational Intelligence (Springer); Fellow, Center of Excellence in Terrorism, Resilience, Intelligence & Organized Crime Research (CENTRIC).*
- *Prof. Dr. Juan-Vicente Capella-Hernandez; Universitat Politecnica de Valencia (UPV), Department of Computer Engineering (DISCA), Valencia, Spain*
- *Prof. Emeritus Kevin Daimi (Steering Committee); Department of Mathematics, Computer Science and Software Engineering, University of Detroit Mercy, Detroit, Michigan, USA*
- *Prof. Leonidas Deligiannidis (Steering Committee); Department of Computer Information Systems, Wentworth Institute of Technology, Boston, Massachusetts, USA*
- *Prof. Mary Mehrnoosh Eshaghian-Wilner (Steering Committee); Professor of Engineering Practice, University of Southern California, California, USA; Adjunct Professor, Electrical Engineering, University of California Los Angeles, Los Angeles (UCLA), California, USA*
- *Prof. Louie Lolong Lacatan; Chairperson, Computer Engineering Department, College of Engineering, Adamson University, Manila, Philippines; Senior Member, International Association of Computer Science and Information Technology (IACSIT), Singapore; Member, International Association of Online Engineering (IAOE), Austria*
- *Prof. Hyo Jong Lee; Director, Center for Advanced Image and Information Technology, Division of Computer Science and Engineering, Chonbuk National University, South Korea*

- *Dr. Housseem Eddine Nouri; Informatics Applied in Management, Institut Supérieur de Gestion de Tunis, University of Tunis, Tunisia*
- *Prof. Dr., Eng. Robert Ehimen Okonigene (Steering Committee); Department of Electrical & Electronics Engineering, Faculty of Engineering and Technology, Ambrose Alli University, Edo State, Nigeria*
- *Ashu M. G. Solo (Publicity), Fellow of British Computer Society, Principal/R&D Engineer, Maverick Technologies America Inc.*
- *Prof. Fernando G. Tinetti (Steering Committee); School of Computer Science, Universidad Nacional de La Plata, La Plata, Argentina; also at Comision Investigaciones Cientificas de la Prov. de Bs. As., Argentina*
- *Prof. Layne T. Watson (Steering Committee); Fellow of IEEE; Fellow of The National Institute of Aerospace; Professor of Computer Science, Mathematics, and Aerospace and Ocean Engineering, Virginia Polytechnic Institute & State University, Blacksburg, Virginia, USA*
- *Prof. Jane You (Steering Committee); Associate Head, Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong*
- *Dr. Farhana H. Zulkernine; Coordinator of the Cognitive Science Program, School of Computing, Queen's University, Kingston, ON, Canada*

Contents

Part I Military and Defense Modeling and Simulation

Julia and Singularity for High Performance Computing	3
Joseph Tippit, Douglas D. Hodson, and Michael R. Grimaila	
Trojan Banker Simulation Utilizing Python	17
Drew Campbell, Jake Hall, Iyanuoluwa Odebode, Douglas D. Hodson, and Michael R. Grimaila	
CovidLock Attack Simulation	25
Amber Modlin, Andrew Gregory, Iyanuoluwa Odebode, Douglas D. Hodson, and Michael R. Grimaila	
The New Office Threat: A Simulation of Watering Hole Cyberattacks	35
Braeden Bowen, Jeremy Eraybar, Iyanuoluwa Odebode, Douglas D. Hodson, and Michael R. Grimaila	
Simulation of SYN Flood Attack and Counter-Attack Methods Using Average Connection Times	43
Hai Vo, Raymond Kozlowski, Iyanuoluwa Odebode, Douglas D. Hodson, and Michael R. Grimaila	

Part II Computational Intelligence, Data Science, HPC, Optimization and Applications

Dielectric Polymer Genome: Integrating Valence-Aware Polarizable Reactive Force Fields and Machine Learning	51
Kuang Liu, Antonina L. Nazarova, Ankit Mishra, Yingwu Chen, Haichuan Lyu, Longyao Xu, Yue Yin, Qinai Zhao, Rajiv K. Kalia, Aiichiro Nakano, Ken-ichi Nomura, Priya Vashishta, and Pankaj Rajak	
A Methodology to Boost Data Science in the Context of COVID-19	65
Carlos J. Costa and Joao Tiago Aparicio	

Shallow SqueezeNext Architecture Implementation on BlueBox2.0 77
 Jayan Kant Duggal and Mohamed El-Sharkawy

Dark Data: Managing Cybersecurity Challenges and Generating Benefits 91
 Haydar Teymourlouei and Lethia Jackson

Implementing Modern Security Solutions for Challenges Faced by Businesses in the Internet of Things (IoT) 105
 Haydar Teymourlouei and Daryl Stone

Trusted Reviews: Applying Blockchain Technology to Achieve Trusted Reviewing System 119
 Areej Alhogail, Ghadah Alhudhayf, Jood Alanzy, Jude Altalhi, Shahad Alghunaim, and Shahad Alnasser

Large-Scale Parallelization of Lattice QCD on Sunway TaihuLight Supercomputer 133
 Ailin Xu, Zhongzhi Luan, Ming Gong, and Xiangyu Jiang

Part III Scientific Computing, Modeling and Simulation

Reverse Threat Modeling: A Systematic Threat Identification Method for Deployed Vehicles 151
 Mona Gierl, Reiner Kriesten, Peter Neugebauer, and Eric Sax

PRNG-Broker: A High-Performance Broker to Supply Parallel Streams of Pseudorandom Numbers for Large-Scale Simulations 167
 Andre Pereira and Alberto Proenca

Numerical Modeling of a Viscous Incompressible Fluid Flow in a Channel with a Step 185
 Saeed M. Dubas, Paul Bouthellier, Nihal Siriwardana, and Laura Wieserman

Modeling, Simulation, and Verification for Structural Stability and Vibration Reduction of Gantry Robots for Shipyard Welding Automation Using ANSYS Workbench® and Recurdyn® 201
 Seung Min Bae, Won Jee Chung, Hui Geon Hwang, and Yeon Joo Ahn

Long Short-Term Memory Neural Network on the Trajectory Computing of Direct Dynamics Simulation 217
 Fred Wu, Tejaswi Jonnalagadda, Colmenares-diaz Eduardo, Sailaja Peruka, Poojitha Chapala, and Pooja Sonmale

Evaluating the Effect of Compensators and Load Model on Performance of Renewable and Nonrenewable DGs 235
 H. Shayeghi, H. A. Shayanfar, and M. Alilou

The Caloric Curve of Polymers from the Adaptive Tempering Monte Carlo Method 247
 Greg Helmick, Yoseph Abere, and Estela Blaisten-Barojas

Part IV Scientific Computing, Computational Science, and Applications

A New Technique of Invariant Statistical Embedding and Averaging in Terms of Pivots for Improvement of Statistical Decisions Under Parametric Uncertainty 257
 Nicholas A. Nechval, Gundars Berzinsh, and Konstantin N. Nechval

A Note on the Sensitivity of Generic Approximate Sparse Pseudoinverse Matrix for Solving Linear Least Squares Problems 275
 A. D. Lipitakis, G. A. Gravvanis, C. K. Filelis-Papadopoulos, and D. Anagnostopoulos

Undergraduate Research: Bladerunner 293
 Adina Paddy, Cha Xiong, Colt Henderson, Tuu Le, and Daren Wilcox

Comparison of the IaaS Security Available from the Top Three Cloud Providers 307
 L. Kate Tomchik

Orientation and Line Thickness Determination in Binary Images 325
 Sean Matz

Greedy Navigational Cores in the Human Brain 337
 Zalán Heszberger, András Majdán, András Biró, András Gulyás, László Balázs, Vilmos Németh, and József Bíró

A Multicommodity Flow Formulation and Edge Exchange Heuristic Embedded in Cross Decomposition for Solving Capacitated Minimum Spanning Tree Problem 347
 Han-Suk Sohn and Dennis Bricker

Elemental Analysis of Oil Paints 363
 Shijun Tang, Rosemarie C. Chinni, Amber Malloy, and Megan Olsson

Part V High-Performance Computing, Parallel and Distributed Processing

Toward a Numerically Robust and Efficient Implicit Integration Scheme for Parallel Power Grid Dynamic Simulation Development in GridPACK™ 371
 Shuangshuang Jin, Shrirang G Abhyankar, Bruce J Palmer, Renke Huang, William A Perkins, and Yousu Chen

Improving Analysis in SPMD Applications for Performance Prediction 387
Felipe Tirado, Alvaro Wong, Dolores Rexachs, and Emilio Luque

Directive-Based Hybrid Parallel Power System Dynamic Simulation on Multi-core CPU and Many-Core GPU Architecture..... 405
Cong Wang, Shuangshuang Jin, and Yousu Chen

Parallel Computation of Gröbner Bases on a Graphics Processing Unit 417
Mark Hojnacki, Andrew Leeseberg, Jack O’Shaughnessy, Michael Dauchy, Alan Hylton, Leah Gold, and Janche Sang

Single Core vs. Parallel Software Algorithms on a Multi-core RISC Processor..... 433
Austin White and Michael Galloway

MPI Communication Performance in a Heterogeneous Environment with Raspberry Pi..... 451
Oscar C. Valderrama Riveros and Fernando G. Tinetti

A FPGA-Based Heterogeneous Implementation of NTRUEncrypt 461
Hexuan Yu, Chaoyu Zhang, and Hai Jiang

High-Performance and Energy-Efficient FPGA-GPU-CPU Heterogeneous System Implementation 477
Chaoyu Zhang, Hexuan Yu, Yuchen Zhou, and Hai Jiang

Preliminary Performance and Programmability Comparison of the Thick Control Flow Architecture and Current Multicore CPUs 493
Martti Forsell, Sara Nikula, and Jussi Roivainen

Part VI Communication Strategies, Internet Computing, Cloud, and Computational Science

Refactor Business Process Models with Redundancy Elimination 509
Fei Dai, Huihui Xue, Zhenping Qiang, Lianyong Qi, Mohammad R. Khosravi, and Zhihong Liang

A Shortest-Path Routing Algorithm in Bicubes 525
Masaaki Okada and Keiichi Kaneko

An NPGA-II-Based Multi-objective Edge Server Placement Strategy for IoV 541
Xuan Yan, Zhanyang Xu, Mohammad R. Khosravi, Lianyong Qi, and Xiaolong Xu

Automatic Mapping of a Physical Model into a Conceptual Model for a NoSQL Database 557
 Fatma Abdelhedi, Amal Ait Brahim, Rabah Tighilt Ferhat, and Gilles Zurfluh

Composition of Parent–Child Cyberattack Models 579
 Katia P. Maxwell, Mikel D. Petty, C. Daniel Colvett, Tymaine S. Whitaker, and Walter A. Cantrell

Tree-Based Fixed Data Transmission for Healthcare Sensor Networks 593
 Susumu Shibusawa and Toshiya Watanabe

Survey on Recent Active Learning Methods for Deep Learning..... 609
 Azar Alizadeh, Pooya Tavallali, Mohammad R. Khosravi, and Mukesh Singhal

Cloud-Edge Centric Service Provisioning in Smart City Using Internet of Things 619
 Manoj Kumar Patra, Sampa Sahoo, Bibhudatta Sahoo, and Ashok Kumar Turuk

Challenges for Swarm of UAV-Based Intelligence 633
 Muhammed Akif Ağca, Peiman Alipour Sarvari, Sébastien Faye, and Djamel Khadraoui

Contrived and Remediated GPU Thread Divergence Using a Flattening Technique 647
 Lucas Vespa and Genevieve Peters

Prototype of MANET Network with Ring Topology for Mobile Devices..... 659
 Ramses Fuentes Pérez, Erika Hernández Rubio, Diego D. Flores Nogueira, and Amilcar Meneses Viveros

Part VII International Workshop

New State-of-the-Art Results on ESA’s Messenger Space Mission Benchmark..... 669
 Martin Schlueter, Mohamed Wahib, and Masaharu Munetomo

Crawling Low Appearance Frequency Character Images for Early-Modern Japanese Printed Character Recognition..... 683
 Nanami Fujisaki, Yu Ishikawa, Masami Takata, and Kazuki Joe

Application of the Orthogonal QD Algorithm with Shift to Singular Value Decomposition for Large Sparse Matrices 697
 Hiroki Tanaka, Taiki Kimura, Tetsuaki Matsunawa, Shoji Mimotogi, Masami Takata, Kinji Kimura, and Yoshimasa Nakamura

On an Implementation of the One-Sided Jacobi Method with High Accuracy	713
Masami Takata, Sho Araki, Kinji Kimura, and Yoshimasa Nakamura	
Improvement of Island Genetic Algorithm Using Multiple Fitness Functions	725
Shigeka Nakajima and Masami Takata	
High-Performance Cloud Computing for Exhaustive Protein-Protein Docking	737
Masahito Ohue, Kento Aoyama, and Yutaka Akiyama	
HoloMol: Protein and Ligand Visualization System for Drug Discovery with Augmented Reality	747
Atsushi Koyama, Shingo Kawata, Wataru Sakamoto, Nobuaki Yasuo, and Masakazu Sekijima	
Leave-One-Element-Out Cross-Validation for Band Gap Prediction of Halide Double Perovskites	759
Hiroki Igarashi, Nobuaki Yasuo, and Makasazu Sekijima	
Interpretation of ResNet by Visualization of the Preferred Stimulus in Receptive Fields	769
Genta Kobayashi and Hayaru Shouno	
Bayesian Sparse Covariance Structure Analysis for Correlated Count Data	781
Sho Ichigozaki, Takahiro Kawashima, and Hayaru Shouno	
Gaze Analysis of Modification Policy in Debugging an Embedded System	793
Takeru Baba, Erina Makihara, Hirotaka Yoneda, Kiyoshi Kiyokawa, and Keiko Ono	
Part VIII Simulation and Modeling	
Modern Control Methods of Time-Delay Control Systems	811
R. Bars, Cs. Bányász, and L. Keviczky	
An Interactive Software to Learn Pathophysiology with 3D Virtual Models	825
Abel A. Reyes, Youxin Luo, Parashar Dhakal, Julia Rogers, Manisa Baker, and Xiaoli Yang	
A Simulation-Optimization Technique for Service Level Analysis in Conjunction with Reorder Point Estimation and Lead-Time Consideration: A Case Study in Sea Port	839
Mohammad Arani, Saeed Abdolmaleki, Maryam Maleki, Mohsen Momenitabar, and Xian Liu	

Sustainability, Big Data, and Local Community: A Simulation Case Study of a Growing Higher Education Institution 859
 Anatoly Kurkovsky

Vehicle Test Rig Modeling and Simulation 873
 Sara Boyle

Modelling and Simulation of MEMS Gyroscope with Coventor MEMS+ and MATLAB/Simulink Software 881
 Jacek Nazdrowicz, Adam Stawinski, and Andrzej Napieralski

Ground Vehicle Suspension Optimization Using Surrogate Modeling ... 887
 Jeremy Mange

Part IX Modeling, Visualization, Computational Science, and Applications

Enhanced Freehand Interaction by Combining Vision and EMG-Based Systems in Mixed-Reality Environments 895
 Carol Naranjo-Valero, Sriram Srinivasa, Achim Ebert, and Bernd Hamann

Parameterizations of Closed-Loop Control Systems would be perfectly fine 911
 Cs. Bányász, L. Keviczky, and R. Bars

A Virtual Serious Game for Nursing Education 927
 Youxin Luo, Abel A. Reyes, Parashar Dhakal, Manisa Baker, Julia Rogers, and Xiaoli Yang

Modeling Digital Business Strategy During Crisis 943
 Sakir Yucel

Dealing Bridge Hands: A Study in Random Data Generation 961
 Peter M. Maurer

An Empirical Study of the Effect of Reducing Matching Frequency in High-Level Architecture Data Distribution Management 975
 Mikel D. Petty

Research on Repair Strategy of Heterogeneous Combat Network 991
 Yanyan Chen, Yonggang Li, Shangwei Luo, and Zhizhong Zhang

The Influence of Decorations and Word Appearances on the Relative Size Judgment in Viewers of Tag Clouds 1005
 Khaldoun Dhou, Robert Kosara, Mirsad Hadzikadic, and Mark Faust

Automation of an Off-Grid Vertical Farming System to Optimize Power Consumption 1015
 Otto Randolph and Bahram Asiabanpour

Workflow for Investigating Thermodynamic, Structural, and Energy Properties of Condensed Polymer Systems 1023
 James Andrews and Estela Blaisten-Barojas

Part X Grid, Cloud, & Cluster Computing – Methodologies and Applications

The SURF System for Continuous Data and Applications Placement Across Clouds..... 1033
 Oded Shmueli and Itai Shaked

The Abaco Platform: A Performance and Scalability Study on the Jetstream Cloud..... 1059
 Christian R. Garcia, Joe Stubbs, Julia Looney, Anagha Jamthe, Mike Packard, and Kreshel Nguyen

Enterprise Backend as a Service (EBaaS) 1077
 Gokay Saldamli, Aditya Doshatti, Darshil Kapadia, Devashish Nyati, Maulin Bodiwala, and Levent Ertaul

Secure Business Intelligence 1101
 Aspen Olmsted

Framework for Monitoring the User’s Behavior and Computing the User’s Trust 1119
 Maryam Alruwaythi and Kendall Nygard

Selective Compression Method for High-Quality DaaS (Desktop as a Service) on Mobile Environments 1137
 Baikjun Choi and Sooyong Park

SURF: Optimized Data Distribution Technology..... 1149
 Oded Shmueli and Itai Shaked

Securing Mobile Cloud Computing Using Encrypted Biometric Authentication 1177
 Iehab AlRassan

Performance Analysis of Remote Desktop Session Host with Video Playback Scenarios 1189
 Baikjun Choi and Sooyong Park

Mining_RNA: WEB-Based System Using e-Science for Transcriptomic Data Mining	1195
Carlos Renan Moreira, Christina Pacheco, Marcos Vinícius Pereira Diógenes, Pedro Victor Morais Batista, Pedro Fernandes Ribeiro Neto, Adriano Gomes da Silva, Stela Mirla da Silva Felipe, Vânia Marilande Ceccatto, Raquel Martins de Freitas, Thalia Katiane Sampaio Gurgel, Exlley Clemente dos Santos, Cynthia Moreira Maia, Thiago Alefy Almeida e Sousa, and Cícilia Raquel Maia Leite	
Index	1205

Part I
Military and Defense Modeling and
Simulation

Julia and Singularity for High Performance Computing



Joseph Tippit, Douglas D. Hodson, and Michael R. Grimaila

1 Introduction

Our research team is focusing on developing a software suite of tools to simulate quantum systems, specifically in regard to quantum teleportation. Our goal is to create a library general enough for researchers to be able to apply our software to many different quantum problems rather than one specific one and to keep it as open source and distributable as possible. Due to the high level of computation needed to fully model these systems, code run-times can easily and exponentially be driven upward as the matrices involved become increasingly larger.

High-level dynamic languages such as Python, despite their benefits in ease of use and readability, simply do not offer the speed we require. However, lower-level languages such as C offer considerably less flexibility and greater difficulty in developing and maintaining code. As a middle ground to this, we have chosen to use the relatively newer programming language Julia. Julia, while being a dynamic language, was developed with speed in mind, targeting researchers and data scientists hoping to get as much performance out of their code as possible while still maintaining inherent readability and ease of use. This made it an obvious first choice for our research.

Also, due to the high level of matrix calculations involved in quantum mechanics, our research can benefit greatly from the performance gains offered by running as much of our code as possible on GPUs rather than the traditional CPU. Julia offers an extensive amount of support in this area, as it has native GPU programming capabilities offered by the `CUDAnative.jl` library, as well as multiple other libraries

J. Tippit (✉) · D. D. Hodson · M. R. Grimaila
Air Force Institute of Technology, Wright-Patterson AFB, Dayton, OH, USA
e-mail: joseph.tippit@afit.edu; douglas.hodson@afit.edu; michael.grimaila@afit.edu

for support. Combined with Julia’s just-in-time compiler, these libraries offer a great level of efficiency in the kernel launch sequence.

In keeping distributability in mind, we have also opted to develop our library inside of the container engine Singularity. Collaborators can be limited by local security practices and administrative privileges needed to install dependencies required to run the software and code of others. This is combined with the need to maintain version control of software libraries fundamental to their own workflow. Containerization as a technology has risen to meet these needs. Containers offer similar benefits to virtual machines such as managing library dependencies and running multiple isolated operating systems (OS) from the same machine. They do, however, have certain advantages more critical to our work.

Specifically, Singularity makes use of a definition file, where software such as the operating system and essential libraries are defined. This allows us to share our work with others, ensuring all versions and libraries will exactly match our own without interfering with their workflow. This also offers a kind of version control for our work and makes it easier to develop on one machine and execute on another. All that is required is to simply build the container from the definition file, and everything that is required will be installed without having to worry about system administration. Companies such as NVIDIA also offer large repositories of containers pre-built to meet many different needs, allowing us and other researchers to further focus on our research.

Another key benefit we see in using containers is their ability to directly share the kernel of the host OS without the need for a hypervisor, defined later, as is required for a virtual machine. This allows them to directly access the resources of a physical machine with minimal overhead. This is crucial, as much of our code will revolve around utilization of GPUs and getting as much of their speedup as possible. Combined with the definition file, containers only need to install what is absolutely essential to our workflow, sharing everything else with the host OS. This is in contrast to virtual machines, which need to install and run a full OS, requiring more overhead.

Singularity has also been developed with researchers in mind, assuming no administrative privileges and targeting high-performance computing. These reasons have made it ideal for our research. Throughout the course of this paper, we will provide further justification for why we chose both Singularity and Julia as fundamental tools to our work.

2 The Julia Programming Language

Traditionally, high-level dynamic languages have lagged behind lower-level static languages in terms of performance. The emphasis on readability, ease of use, and productivity is believed to come at the cost of run-times and execution speeds. Prototyping is thus done in a high-level language and then fully implemented in a low-level language for speed.

With Julia, this is not the case. From its initial development, the Julia Programming Language was designed with speed in mind while still offering the same productive programming of a high-level language. In a blog post about why they created it, Julia’s developers are quoted as saying, “We want the speed of C with the dynamism of Ruby. We want a language that’s homoiconic, with true macros like Lisp, but with obvious, familiar mathematical notation like Matlab. We want something as usable for general programming as Python, as easy for statistics as R, as natural for string processing as Perl, as powerful for linear algebra as Matlab, as good at gluing programs together as the shell” [1].

By examining Julia’s Pythonic syntax with natural mathematical notation, as well as the micro-benchmarks in Figs. 1 and 2, we can easily see that they have achieved their goals. We will further examine three key features that lend themselves to this success:

- Julia’s just-in-time compiler and type system
- Low-level virtual machine (LLVM)
- Native GPU support

2.1 Just-in-Time Compilation

Just-in-time (JIT) compilation is a technique that converts high-level languages into machine code executable directly on the CPU when it is run [3]. This means that Julia, unlike Python, is a compiled language. This is a benefit for its speed, as

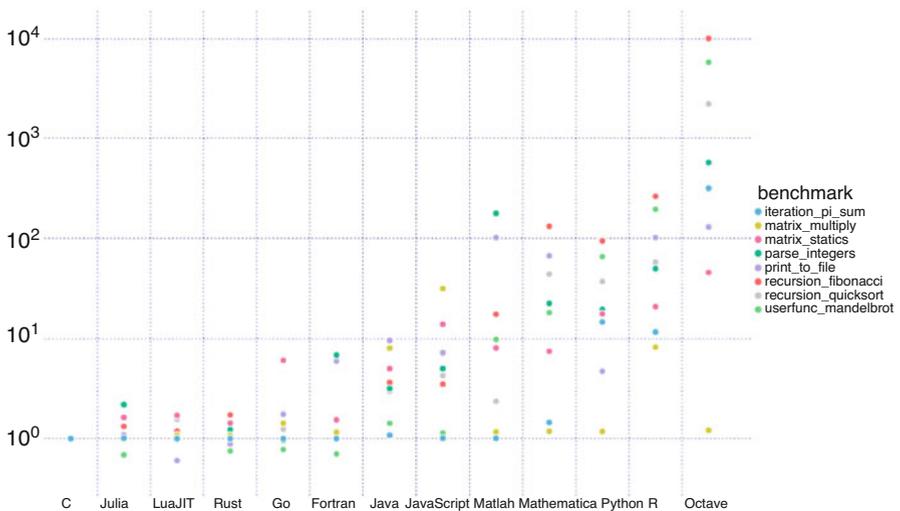


Fig. 1 Julia Micro-Benchmarks [2]

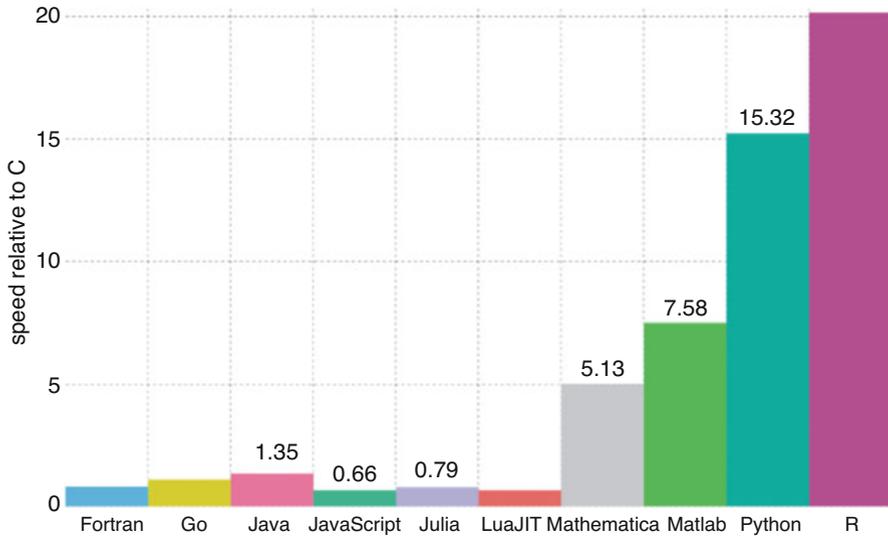


Fig. 2 Language speedups relative to C [3]

it drops the overhead of interpretation. The difference between languages such as C, however, is that the code is compiled at run-time rather than beforehand. To adequately benchmark Julia run-times, it is therefore necessary to run the code once so that it is compiled and then a second time to benchmark. Otherwise, you will also time the overhead of compiling the code as well.

We can see the steps involved in the Julia compilation process in Fig. 3. Julia also offers macros to allow the programmer to see the output of each step. For example, if one is interested in seeing the LLVM bitcode output, the `@code_llvm` macro could be placed in front of the desired function. If, instead, one wanted to see the actual assembly that Julia compiled to, use the `@code_native` instead. These macros can give the programmer insight into where optimizations in their code can be made.

The speedups offered by Julia are also largely due to multiple dispatch. Essentially, this is a method whereby multiple functions are automatically created to perform the same operation, and one is selected based on the types of the individual inputs. This allows the compiler to make certain optimizations and improvements to each function without the user needing to manually define the same operation for every combination of possible input types.

Adding two numbers together is an excellent example of this. On the computer hardware, the value 1.0 is stored uniquely different from 1. The former is stored as a float and the latter as an integer. Summing these together would output 2.0, a float. If instead both had been kept integers, the resulting output would have also been an integer.

A central idea behind multiple dispatch is type stability. Type stability is the concept that the type of a return value depends only on the types of the individual

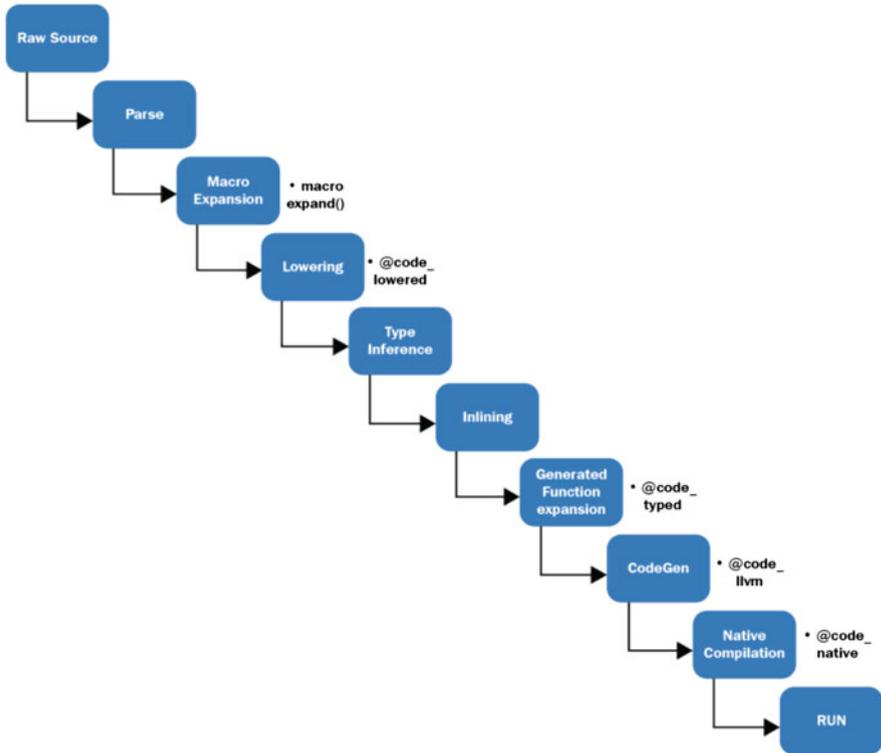


Fig. 3 Julia Compilation Process [3]

inputs and not on their values. It is, ultimately, what makes multiple dispatch work by allowing the compiler to choose the correct function to perform the desired operation. The book Julia High Performance: Optimizations, Distributed Computing, Multithreading, and GPU Programming with Julia 1.0 and Beyond, 2nd Edition by Avik Sengupta has a simple example of this highlighted in the code examples below [3]:

```
function pos(x)
    if x < 0
        return 0
    else
        return x
    end
end
```

We can see that inputting float 2.5 will return the float 2.5. However, -2.5 will return 0 as an integer. This is type instability. Julia offers many ways to identify and fix type instability; however, Julia’s compiler has been optimized to make even code

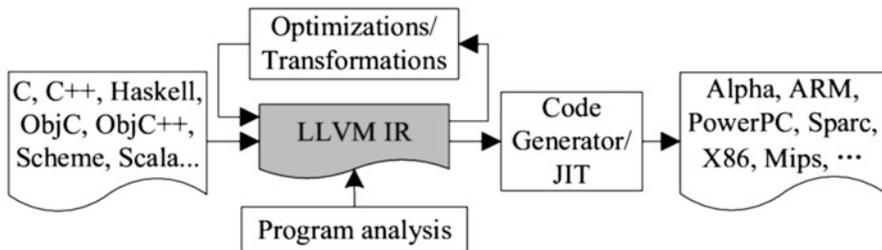


Fig. 4 High level LLVM overview [4]

with type instabilities execute almost as fast as code without. The ability to focus more on writing our code while letting the compiler worry about the lower-level optimizations is a huge benefit to our work.

2.2 *LLVM*

Prior to being passed off to the JIT compiler, Julia uses the compiler infrastructure LLVM to convert its syntax into an intermediate representation in memory. This syntax allows LLVM to make different optimizations before being compiled to machine code [5]. These optimizations are implemented as passes in LLVM, and there are three different categories: analysis, transform, and utility passes [6].

In the analysis passes, information is collected for other passes to use in regard to debugging and program visualization, and transform passes will all mutate the program in some way. Utility passes is a categorical catchall for passes that have some utility but do not fit into the other two [6]. The final output is the intermediate representation, which is a low-level language similar to assembly. As previously mentioned, it can be viewed by adding the `@code_llvm` macro to a function. Figure 4 shows a high-level overview of this process.

2.3 *Native GPU Support*

One of the most influential reasons in using Julia for our research is its native GPU support. Julia has multiple libraries for utilizing the GPU at differing levels of abstraction, as shown in Fig. 5. As we will be using NVIDIA GPUs, CUDA support is of the greatest interest to us. Using the `CUDAnative.jl` library, in conjunction with `CUDAadv.jl` and `CuArrays.jl`, we are capable of doing the same low-level GPU programming that could be done in CUDA C++. These libraries provide a means of interfacing with the CUDA driver and run-time libraries, writing kernels, and managing execution [7].

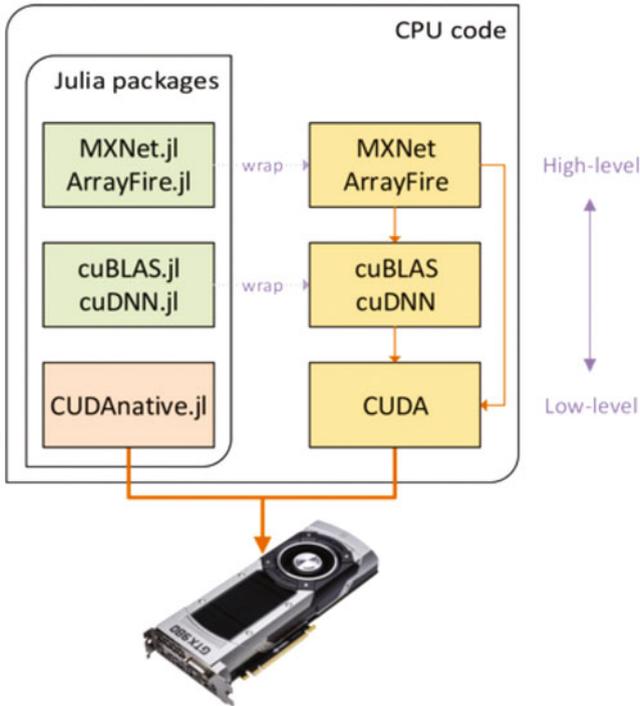


Fig. 5 Julia GPU libraries at differing levels of abstraction [7]

These libraries integrate into Julia’s JIT compiler, allowing the code to be compiled directly to GPU assembly. Ultimately, what this means is that you can use Julia for the GPU almost exactly how you would for the CPU. This gives the programmer the same productivity Julia offers and combines it with the inherent parallelism of the GPU, generating efficient PTX code [7]. Similar to how we can view the LLVM intermediate representation and native assembly code, we can also view the PTX generated by the LLVM PTX compiler backend using the `@device_code_sass` macro (Fig. 6).

Additionally, the Julia team has been porting the Rodinia, a benchmark suit for heterogeneous computing, to Julia. The Rodinia benchmark suite works by measuring parallel communication patterns, synchronization techniques, as well as power consumption in order to provide a standard benchmark to compare platforms [8]. We can see how Julia compares against CUDA C++ in Fig. 7.

CUDAnative.jl can be considered the basic library essential to using CUDA in Julia; however, CuArrays.jl is arguably the most significant. GPU code must be vectorized in order to gain any performance increases, and the array is the fundamental type for this. The CuArrays data type allows arrays to be created for use on the GPU just like any other array in Julia [3]. Overall, these libraries make Julia just as useful and powerful for programming on the GPU as C++ while still keeping

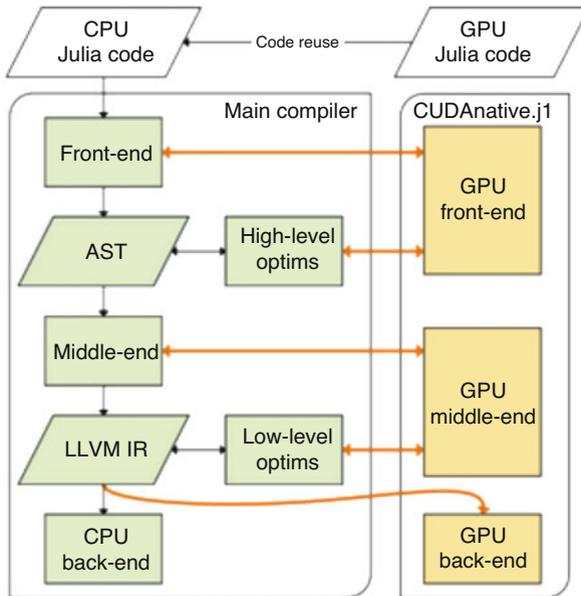


Fig. 6 Overview of the CUDA native.jl compiler [7]

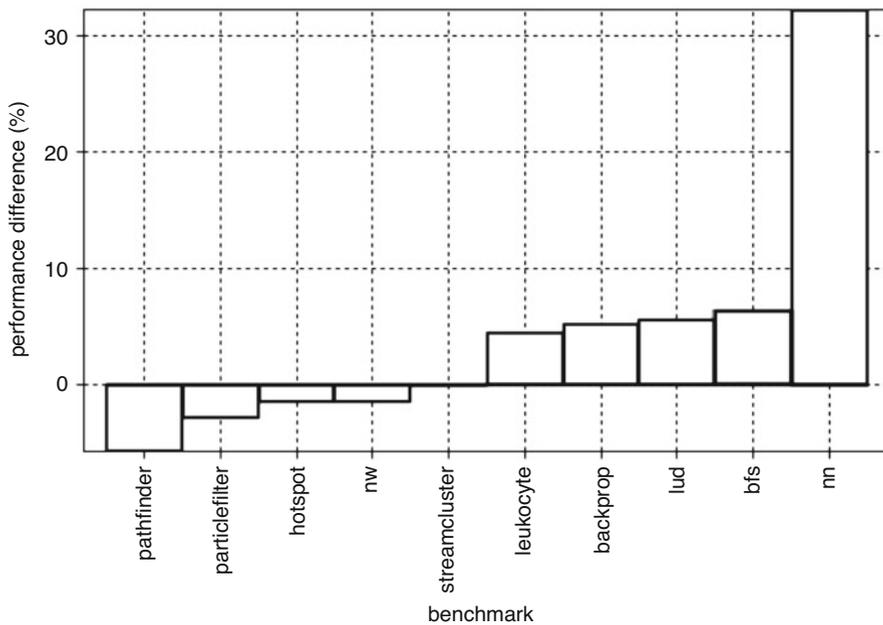


Fig. 7 Performance difference between CUDA C++ and CUDA native.jl using the Rodinia benchmark [7]

in line with a high-level language paradigm. Intuitive mathematical syntax, a highly optimized compiler, and the ability to make low-level GPU abstractions have given Julia a competitive edge in high-performance computing as well as making it the decisive choice for our research.

3 Containerization

3.1 *Differences Between Containers and Virtual Machines*

Virtualization developed as a means of meeting the demands of shared resources among a large collection of users [9]. Universities and companies alike utilize virtual machines (VMs) to provide their students and employees with a way of accessing the organization's computer resources remotely from a shared server. This can reduce the costs of having to provide users these same resources physically.

Conceptually, virtualization is an abstraction of the hardware and resources of a physical machine from the operating systems and software running in a virtual machine. Fundamental to this is the software called the hypervisor. A hypervisor, or virtual machine monitor (VMM), isolates the operating system and resources on the physical machine (the host) from the virtual machine (the guest) and oversees all VMs on the host. Users can create many VMs on one machine, allocating different amounts of resources such as memory and storage to each. The hypervisor sees these resources as a pool and manages their utilization among every running VM as needed [10].

Essentially, the virtual machine exists on the host as a folder. This allows it to be easily moved and copied around. The image containing the VM includes a complete operating system and its corresponding applications, allowing for different operating systems and environment regardless of what the host OS is. However, this comes at the price of storage space and having potentially significant overhead in terms of CPU resources and RAM [11].

In contrast to this, containers exist on the host OS as either a file or a collection of files, allowing them to be highly portable and configurable. Since containers share the same kernel as the host OS, they drop the need for a hypervisor and have direct access to system resources without needing to emulate them. We show a high-level view of this difference in Fig. 8.

Containers also come with just the necessary run-time files, dependencies, and software to run the required software. Due to this, they are more lightweight than a VM and will run natively on a Linux operating system while still isolating their applications from the OS [12]. The file system is also isolated and will only mount certain prescribed and user-directed directories from the host OS. Other than these directories, the file system paths and the files they hold will be different from the host (Fig. 9).

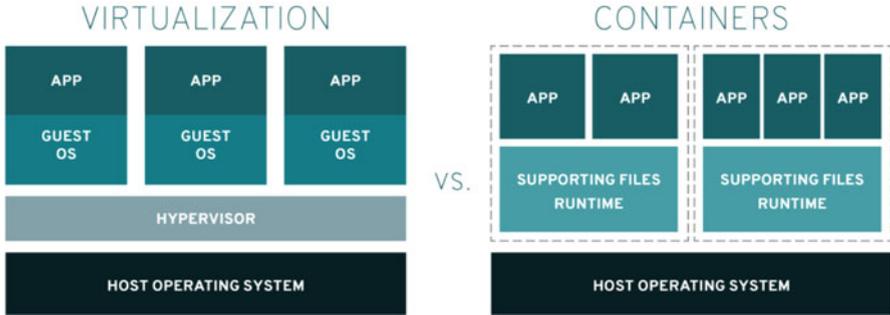


Fig. 8 High-level view of containers vs virtualization [12]

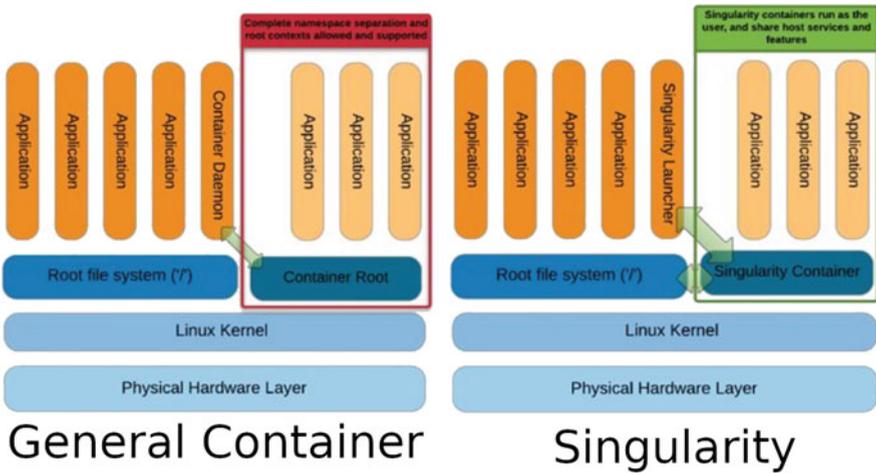


Fig. 9 General Container Architecture compared to Singularity [15]

Both Singularity and Docker, another popular container engine, set up their container environments from a definition file or Dockerfile, respectively. This allows users to maintain a level of version control when sharing their work. These files define the operating system environment and library/software requirements of the application being contained, giving collaborators a straightforward means to ensure all dependencies will be exactly what the original developer intended without needing to have a system administrator install anything. Since containers are also isolated from the host, they will run as just another process and will not interfere with any other workflow or library dependencies.

3.2 *Docker versus Singularity*

Docker is one of the most widely known and used container engines available today. There are a plethora of options available to researchers to immediately pull a container suitable to their needs and begin working. NVIDIA's own GPU Cloud, a hub for high-performance computing container images, hosts their work environments natively as Docker images.

However, Docker was designed to be an enterprise-focused container. No HPC center allows it as it was not intended to support highly distributed parallel applications [13]. Docker also assumes administrative or root privileges and runs applications as such. Not only does this not solve the problem of sharing our work (if collaborators do not have the proper local privileges, they will not be able to run our containers), but this also causes potential security risks as well. Since applications are run as root, any user accessing the container can also be granted the same escalated privileges.

Singularity, on the other hand, assumes no privileges whatsoever and was specifically designed to support HPC and MPI applications. This no trust model makes it ideal for sharing work among researchers and other collaborators, as everything run in the image will be executed with the same privileges as the user. The run-time writes the UID and GID information to the files within the container, so the privileges are the same because the user is the same. Singularity has a slightly different philosophy from Docker, one of "integration over isolation" [14]. This means it supports being able to map more directories from the host operating system into the guest, integrating and embedding it directly into your workflow. Overall, Singularity is a paradigm targeted at scientific workloads to address the core missions of mobility of compute, reproducibility, HPC support, and security [14].

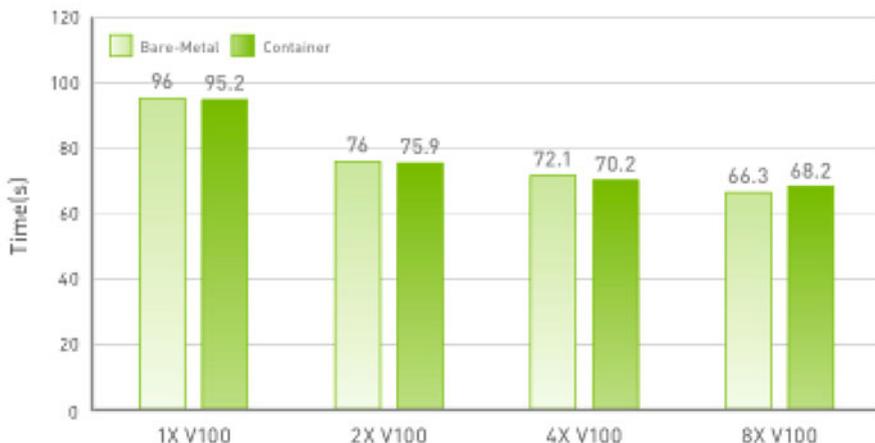
It also supports converting Docker containers directly to Singularity containers in a single command (it will even pull directly from Docker Hub and NVIDIA GPU Cloud, as well as its own Singularity Hub), which means every image on the NVIDIA GPU Cloud will also be supported on Singularity as well. Taken collectively and in keeping in line with other HPC workflows, we have found Singularity to be the best suited container engine for our applications.

3.3 *Singularity and GPUs*

Singularity, by default, makes all host devices accessible to the container. This provides for a seamless integration with GPUs and other devices common to HPC. In fact, Singularity comes stock with command line options to control the usage of GPUs, choosing which to run and when [14].

As previously mentioned, NVIDIA offers many different container options for researchers to pull and immediately begin working. These images come set up with

Containers Performance Similar to Bare-Metal



Intel Xeon E5-2690 v4 | NVIDIA Tesla V100, 16 GB | MILC | sc_15_cluster dataset

Fig. 10 NGC Performance: Containers vs. Bare-Metal [16]

everything one needs to access their GPUs from within a container with similar performance to what one would expect from the host (see Fig. 10 below) and much of NVIDIA’s documentation covers using them from within Singularity [16]. Singularity even has direct access to the host’s driver libraries. These reasons led us to choose Singularity as the container engine of choice for our research.

4 Conclusion

Performance and productivity are two areas crucial to our research. The need to perform massive matrix calculations efficiently has turned us to utilizing GPUs to leverage their inherent parallelism. We have found the programming language Julia, with its highly optimized compiler and native GPU support, to be the ideal basis on which to develop our quantum libraries. Providing us with a high-level language paradigm combined with speeds comparable to C, as well as essentially equivalent support for CUDA as C++, it was a clear choice as a way forward.

However, distributability and reproducibility were two other key factors, leading us to delve into container technology as a solution. Designed with HPC and direct host resource access in mind, Singularity provides us with a means to share our work to the highest degree possible with minimal performance impacts.

References

1. J. Bezanson, S. Karpinski, V.B. Shah, A. Edelman, Why We Created Julia, Feb 2012. [Online]. Available: <https://julialang.org/blog/2012/02/why-we-created-julia/>. Accessed 17 Mar 2020
2. The Julia Project, *Julia 1.3 Documentation*, Aug 2019. [Online]. Available: <https://docs.julialang.org/en/v1/>. Accessed 17 Mar 2020
3. A. Sengupta, *Julia High Performance Computing*, 2nd edn. (Packt Publishing, Birmingham, 2019)
4. J. Zhao, Formalizing the SSA-based compiler for verified advanced program transformations. Ph.D. dissertation, Department of Computer and Information Science, University of Pennsylvania, Philadelphia (2013)
5. R. Lakhanpal, A. Joshi, *Learning Julia* (Packt Publishing, Birmingham, 2017)
6. LLVM Project, *LLVM Documentation*, Mar 2020. [Online]. Available: <https://llvm.org/docs/>. Access 19 Mar 2020
7. T. Besard, High-Performance GPU computing in the Julia programming language, oct 2017. [Online]. Available: <https://devblogs.nvidia.com/gpu-computing-julia-programming-language/>. Accessed 18 Mar 2020
8. G. Honan, S. Shivakumar, A. Siraman, Rodinia Benchmark Suite, University of Pennsylvania, Technical Report, Mar (2017)
9. Oracle, Introduction to Virtualization, Jan 2013. [Online]. Available: <https://docs.oracle.com/>. Accessed 18 Mar 2020
10. Red Hat, What is a Hypervisor? 2020. [Online]. Available: <https://www.redhat.com/en/topics/virtualization/what-is-a-hypervisor>. Accessed 17 Mar 2020
11. A. Strong, Containerization vs. Virtualization: What's the Difference, Nov 2019. [Online]. Available: <https://www.burwood.com/blog-archive/containerization-vs-virtualization>. Accessed 17 Mar 2020
12. Red Hat, What's a Linux Container? 2020. [Online]. Available: <https://www.redhat.com/en/topics/containers/whats-a-linux-container>. Accessed 17 Mar 2020
13. M. Kandes, An introduction to singularity: Containers for scientific and high-performance computing. University of California, San Diego, Technical Report, Feb 2019
14. Sylabs Inc., *Singularity Admin Guide*, 2020. [Online]. Available: <https://sylabs.io/guides/3.5/admin-guides/>. Accessed 19 Mar 2020
15. E. Bollig, Singularity & Containers. University of Minnesota, Technical Report, Nov 2019
16. NVIDIA, Optimized Containers from NVIDIA GPU Cloud, MO, Technical Report (2018)

Trojan Banker Simulation Utilizing Python



Drew Campbell, Jake Hall, Iyanuoluwa Odebode, Douglas D. Hodson, and Michael R. Grimaila

1 Introduction

According to Web safety tips.com, “a Trojan is a type of malware that can effectively hide within your computer system. It can do this by pretending to be something other than what it actually is [1]. Its stealth is part of the reason why it’s so effective at infecting and infiltrating different computer systems.” This makes Trojan’s very dangerous because they are used quite often by hackers to steal data and other information from people without them realizing what is going on [6]. The first Trojan virus was called, “ANIMAL” according to usa.kaspersky.com. ANIMAL was created by a man named John Walker in 1975 [2]. Although the program was not actually a malicious program, it fit within the scope of what a Trojan is considered because it hid another computer program within itself that examined the contents of the user’s computer. A good example of a Trojan is a banking Trojan because it acts as a friendly website or application, but it will attempt to steal credentials from a user when the user accesses the infected website.

In Fig. 1, the attacker has already infected a target website. In this example, the Trojan will enter a user’s system once they access the infected website and click an infected link. The Trojan will be sent to the victim’s computer through malicious

D. Campbell · J. Hall

Wittenberg University, Springfield, OH, USA

e-mail: campbelld4@wittenberg.edu; hallj15@wittenberg.edu

I. Odebode

Oak Ridge Institute for Science & Education, Wright Patterson AFB, Dayton, OH, USA

e-mail: iyanuoluwa.odebode@afit.edu

D. D. Hodson (✉) · M. R. Grimaila

Air Force Institute of Technology, Wright Patterson AFB, Dayton, OH, USA

e-mail: douglas.hodson@afit.edu; michael.grimaila@afit.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_2

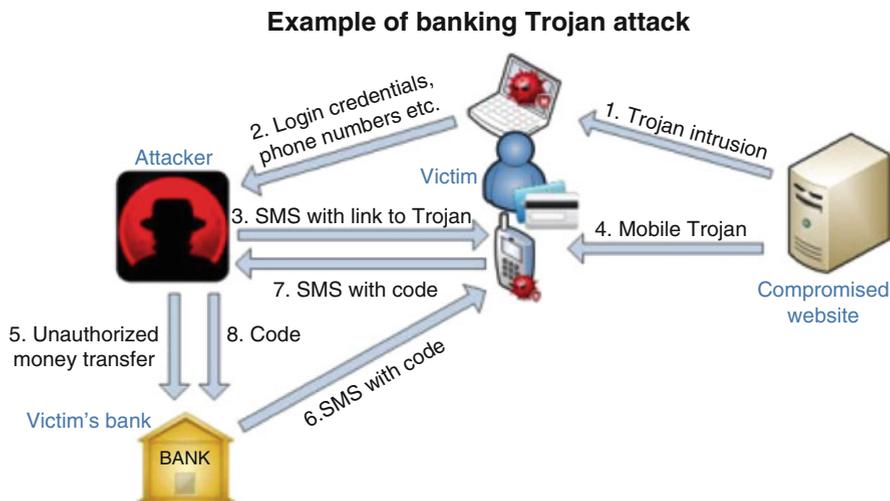


Fig. 1 This diagram depicts how a Trojan Banker operates against a victim computer

packets that have been altered by the attacker (and if this were a standard Trojan, one of the most common ways to attack a user's computer is through social engineering; "social engineering can be traced as far back as the Trojan horse story" (Gray)). Next, a user's log-in credentials along with other personal sensitive information is provided to the attacker through a method of communication between the attacker's computer and the victim's computer. Figure 1 uses an SMS link as one method of communication. This does not notify the user, but it does manage to log the information the victim puts into their computer and relay it back to the attacker. Once an attacker has the credentials needed to access a victim's account, they will log in as the user and withdraw money, steal more data they hadn't gotten from the Trojan yet, or just attempt to damage the victim even more by destroying data. Ultimately, antivirus systems work to detect and remove malicious malware such as Trojans, and those applications usually explain steps that must be taken to avoid obtaining more malware on a user's system.

2 Methodology

This section will detail the approach taken to model a Trojan attack against a series of machines with various levels of security put into place. This section also details the code that was written in the python programming language and will explain the SimPy tool.

In our approach to identifying how long a Trojan can operate on a system without being detected, we decided it was best to identify how a Trojan gains access to a

system, how long that process would take, and how many credentials it can steal before getting removed by the system's security.

The code we created in place of SimPy utilizes a firewall security system that will be repeatedly attacked by a Trojan until the Trojan gains access to a user's system. Then the Trojan will activate and begin to steal credentials while going undetected. Once credentials are stolen, they would be sent back to the attacker's device (The specific credentials stolen are not included in the simulation) [4]. If the Trojan is detected before it manages to steal any information, the Trojan is removed from the target system. If the Trojan manages to go undetected, it will continue to steal any credentials it can until it is detected and removed from the system. This is because we want to simulate how real Trojan malware would operate on a computer system [7].

According to SimPy, "SimPy is a process-based discrete-event simulation framework based on standard Python." SimPy utilizes real-time criteria to manage and run simulations. This allows SimPy to run simulations "as fast as possible" and manages congestion in code by utilizing "yield" statements to pause and resume activities that are occurring in tandem with one another [5]. More specifically, SimPy is a library in the python programming language that uses processes and environments to host event simulations. SimPy monitors the start time and the end time of each simulation as well as monitors the time when events change during a simulation. Therefore, SimPy is able to quickly generate resources that allow a programmer to run reliable simulations [3].

SimPy does not work for the simulation in this paper for three reasons [5]. The first reason why SimPy does not work with this simulation is because in this simulation, a single type of Trojan gains access to a set of machines and collects information until it is caught and removed. Therefore, this simulation operates on a fixed step size, and within the simulation, none of the machines communicate with each other. The only resource that is shared between all the machines are constants (firewall detection chance, log-in chance, and time). "SimPy is overkill for simulations with a fixed step size where your processes don't interact with each other or with shared resources" (SimPy). The second reason why SimPy does not work for this simulation is because SimPy operates all of its simulations on a fixed time which means that all simulations have a set start time and a set end time. For example, the end time in Fig. 2 is after 2 s (`env.run(until=2)`). In our simulation, the Trojan will maintain access to a target system until it is detected and removed. Because of this, the simulation will operate on an unfixed time, therefore rendering SimPy incompatible with the simulation requirements. The last reason why SimPy will not work for our simulation is because SimPy is unable to accommodate for the variables tested to have different settings. In our simulation, each machine has varying levels of security which SimPy cannot distinguish within the simulation.

```

>>> import simpy
>>>
>>> def clock(env, name, tick):
...     while True:
...         print(name, env.now)
...         yield env.timeout(tick)
...
>>> env = simpy.Environment()
>>> env.process(clock(env, 'fast', 0.5))
<Process(clock) object at 0x...>
>>> env.process(clock(env, 'slow', 1))
<Process(clock) object at 0x...>
>>> env.run(until=2)
fast 0
slow 0
fast 0.5
slow 1
fast 1.0
fast 1.5

```

Fig. 2 An example of SimPy simulating two clocks ticking in separate time intervals (SimPy.readthedocs.io)

2.1 Methodology

Our approach to the simulation uses randomly generated numbers to approximate the chances that the Trojan penetrated the firewall, that log-in credentials were logged, and that the antivirus software detected and removed the Trojan. First, global variables and machines were initialized. The global variables were “FirewallPreventionChance,” which is the chance that the firewall would prevent the Trojan from infecting the machines, and “LogonChance,” which is the chance that a user would use their machine to log on to an account and have their credentials stolen. Ten machines were initialized with names (“Machine0,” “Machine1,” etc.) and varied detection chances. The lowest security was set on Machine0 with a 5% chance to detect and remove the Trojan when credentials were logged. The detect chance increased by 10% for each machine; thereafter until the last machine, Machine9 had a 95% detection chance. The simulation would run until all Trojans were detected and removed, so the simulation took place inside a while loop. First, each machine was tested with a different random value against the global FirewallPreventionChance. If the random value was greater than the prevention chance (.1%), the machine was removed from the firewall list and was placed into an infected list, and the time of infection was logged. Once in the infected list, two more random values were created for each machine: a random log-in value and a random

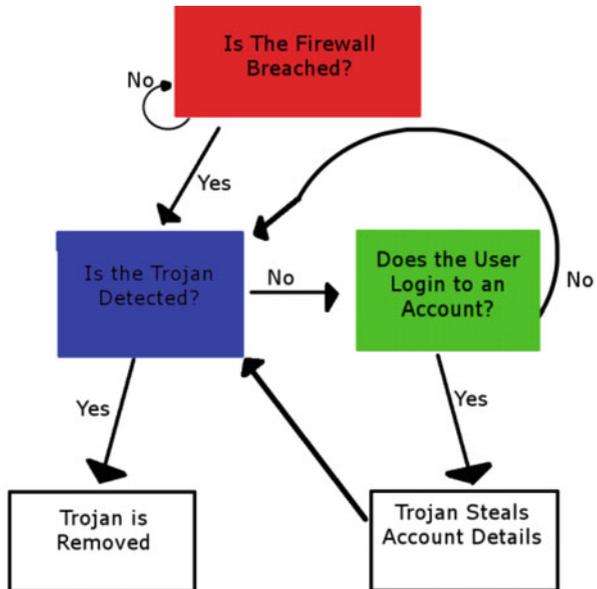
detection value. If the log-in value yielded less than the global LoginChance, the machine logged into an account, and the credentials were stolen by the attacker. If the detection value was less than the computer detection (the 5–95% chance that the machine removes the Trojan), then the time was logged, and the machine was removed from the infected list, indicating that the Trojan had been removed. Once all of the machines had been introduced to the infected list and removed, the simulation ended (Figs. 3 and 4).

3 Results

The results display the amount of data that was stolen and the time that a machine remained infected. The results therefore display how antivirus software and security impact an infected computer system. Three trials were conducted with identical parameters, and results yielded similar trends for each. As expected, the machines with stronger security had fewer credentials stolen and spent less time infected. The Trojan’s were removed more quickly because the increased security had a better chance to catch the malicious software than a machine with less security. We can also see a relationship where the longer a machine was infected, the more credentials were logged.

An important thing discovered in our results is that the level of security does not directly correlate to the amount of time infected or the amount of information stolen. This is a result of using a probability-based simulation. Since the simulation

Fig. 3 The diagram of step-by-step process for the simulation presented in the DEV’s model



Pseudocode for our Simulation:**Variables:**

```
FirewallPreventionChance = 99.9%
```

```
LogonChance = 5%
```

Initialize:

```
#begin the program by initializing the Machine Class #to
create the tested machines with varied names and #security
levels, machine 0 has least security with a #detect chance
of 5% and machine 9 has the most with a #detect chance of
95%.
```

Sim:

```
Initiate 2 lists
```

```
While testing:
```

```
  For i in range(length of list1):
```

```
    Bypass = randomNumber(0.0-100.0)
```

```
    If Bypass > firewallPreventionChance:
```

```
      print( time, 'firewall breached')
```

```
      Move list1[i] to list2
```

```
  For k in range(length of list2):
```

```
    Logon = randomNumber(0-100)
```

```
    Detect = randomNumber(0-100)
```

```
    If logonChance > Logon:
```

```
      print( time, 'a logon on' ,machine number,
            'has been logged')
```

```
    If list2[k].MachineSecurityLevel > Detect:
```

```
      print(time, 'Trojan detected and removed')
```

```
      Remove list2[k] from list2
```

```
  If list1 and list2 are empty:
```

```
    End simulation
```

Fig. 4 Methodology pseudocode

was based on randomly generated numbers and probabilities, there was always a chance that the Trojan did not get detected regardless of the detection chance of the machine. Therefore, some machines with higher security had more time spent infected than some with lower security in some of the trials. The opposite is also true meaning that there was always a chance that the Trojan would be detected. So for the machines with lower security, it was possible for the Trojan to be removed earlier than expected. For example, if we see the results displayed in Fig. 5, we can see that in Trial 1, Machine1 had a much lower infected time than that of Machine2, even though its detection chance was 10% less.

Other attributes tested by our simulation were the methods of infection. These included phishing attacks, untrusted website attacks, and other targeted attacks.

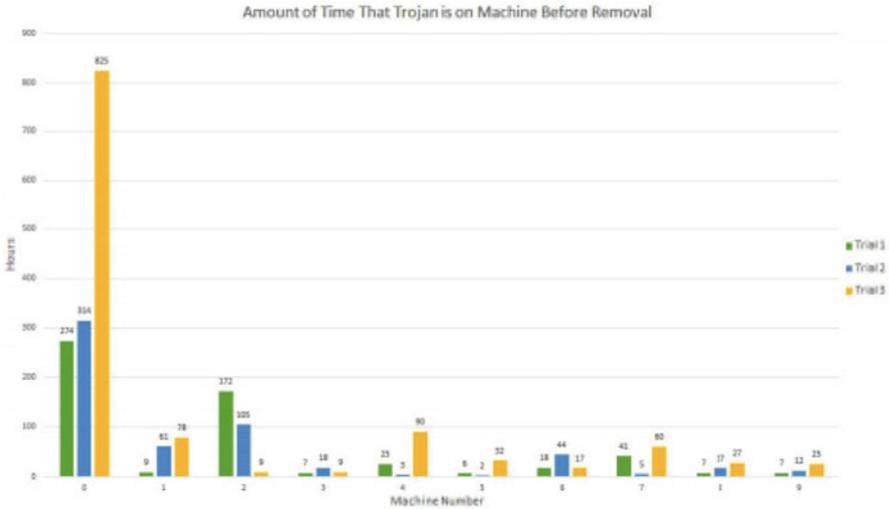


Fig. 5 The amount of time (in hours) that the Trojan malware managed to go undetected by the machine’s security system

Through our research, we found that these methods were the most common ways that Trojans were introduced into a system. What was prevalent in our findings (see Fig. 5) was that social engineering attacks like phishing scams were the overwhelming majority of means to attack. This means that in order for Trojans and other forms of malicious software to infect less computer devices, users need to become more educated on how phishing scams work and how to avoid them (Fig. 6).

4 Conclusion

In conclusion, Trojan malware is very dangerous and works to operate without the user realizing that their system is infected with malicious software. The best ways to ensure a system is not infected with Trojan malware is to keep the most updated patches and security software operational on a device, make sure to use due diligence to identify what is and is not malicious software (learn more about social engineering primarily), and make sure to run security diagnostics on a system often to prevent malicious software from damaging a system.

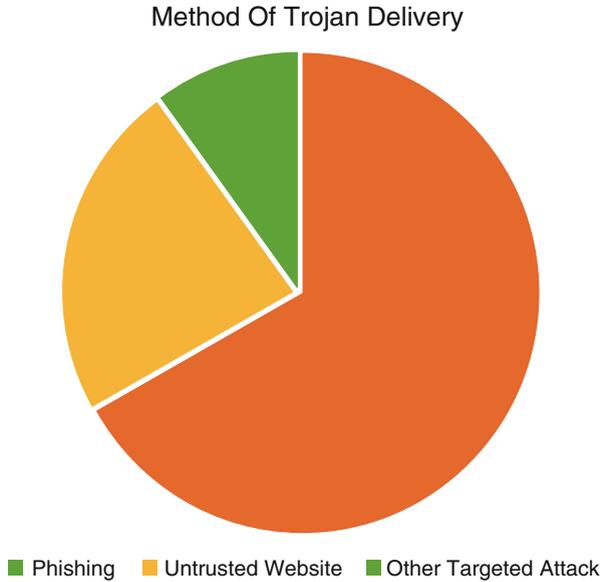


Fig. 6 The amount of time (in hours) that the Trojan malware managed to go undetected by the machine's security system

References

1. D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, B. Sunar, Trojan detection using ic fingerprinting, in *2007 IEEE Symposium on Security and Privacy (SP'07)*. (IEEE, 2007), pp. 296–310
2. J. Kang, Trojan horses of race. *Harv. L. Rev.* **118**, 1489 (2004)
3. N. Matloff, Introduction to discrete-event simulation and the simpy language. *Davis, CA. Dept of Computer Science. University of California at Davis. Retrieved on August 2(2009):1–33* (2008)
4. Trojan horse (computing); Wikipedia, [https://en.wikipedia.org/wiki/Trojan_horse_\(computing\)](https://en.wikipedia.org/wiki/Trojan_horse_(computing)); Access Date: June 15, 2021.
Trojan Computing. Trojan horse virus
5. Team SimPy. Simpy: Discrete event simulation for python. *Python Package Version 3(9)*, 7 (2017)
6. M. Tavallae, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the kdd cup 99 data set, in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications* (IEEE, 2009), pp. 1–6
7. Malware; Wikipedia, <https://en.wikipedia.org/wiki/Malware>; Access Date: June 15, 2021.

CovidLock Attack Simulation



Amber Modlin, Andrew Gregory, Iyanuoluwa Odebode, Douglas D. Hodson, and Michael R. Grimaila

1 Introduction

Amidst the global pandemic over COVID-19, a new form of ransomware has hit Android users. The recently dubbed “CovidLock,” this “lock screen attack” locks users from their phone unless they pay a ransom in cryptocurrency within 2 days [1]. The ransomware accomplishes this feat by asking users access for numerous functions on their Android devices, all under the pretense that it’s a necessity for scanning their area for coronavirus cases. Without the proper encryption code, users are unable to unlock their devices and risk losing their contacts, videos, pictures, and more (Fig. 1).

Ransomware has already been a long-standing problem for Internet-connected devices. These types of attacks are spread out throughout the United States disproportionately among the regions. In the middle of an ongoing crisis, ransomware like CovidLock is troublesome during these times, as users can’t seek help at tech stores to unlock their encrypted devices. For users who lack the technical savvy or proper security updates on their devices, they’re in need of additional aid to not fall into the traps of ransomware (Fig. 2).

A. Modlin · A. Gregory
Wittenberg University, Springfield, OH, USA
e-mail: modlina@wittenberg.edu; gregorya@wittenberg.edu

I. Odebode
Oak Ridge Institute for Science & Education, Wright Patterson AFB, Dayton, OH, USA
e-mail: iyanuoluwa.odebode@afit.edu

D. D. Hodson (✉) · M. R. Grimaila
Air Force Institute of Technology, Wright Patterson AFB, Dayton, OH, USA
e-mail: douglas.hodson@afit.edu; michael.grimaila@afit.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_3

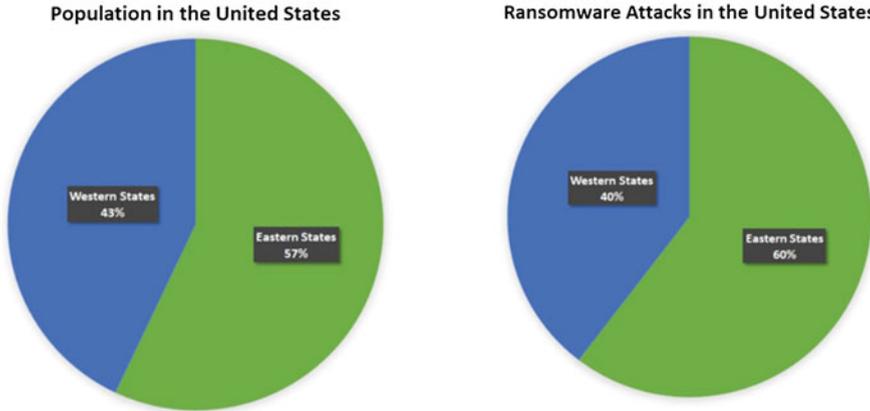


Fig. 1 Comparing population and ransomware attacks in the western and eastern states

State	Population	Ransomware Attacks (2013-Present)	Eastern or Western State
California	39,937,489	21	Western
Texas	29,472,295	32	Western
Florida	21,992,985	16	Eastern
New York	19,440,469	16	Eastern
Pennsylvania	12,820,878	15	Eastern
Illinois	12,659,682	14	Eastern
Ohio	11,747,694	14	Eastern
Georgia	10,736,059	17	Eastern
North Carolina	10,611,862	10	Eastern
Michigan	10,045,029	4	Eastern

Fig. 2 The top ten most populated states and the number of ransomware attacks since 2013

2 Background and Related Information

2.1 Ransomware in the States

While researching the impact ransomware has had in the United States, we ran into the question: “In which states is ransomware more prevalent?” We started this research by sectioning the United States off into two areas, the western states, west of the Mississippi River, and the eastern states, east of the Mississippi River (Fig. 3).

When searching for information on ransomware in the United States, we came across two different interactive maps. Both maps show reported ransomware attacks in the United States, each being constantly updated. One of these maps is a Google Maps called “The Ransomware War by PC Matic.” With a quick glance at this map, we noticed that more attacks were reported in the western states [2]. We wondered why this was the case; was it based on the population of those states, the amount of

State	Population	Ransomware Attacks (2013-Present)	Eastern or Western State
Wyoming	567,025	1	Western
Vermont	628,061	1	Eastern
Alaska	734,002	2	Western
North Dakota	761,723	1	Western
South Dakota	903,027	1	Western
Delaware	982,895	0	Eastern
Rhode Island	1,056,161	7	Eastern
Montana	1,086,759	6	Western
Maine	1,345,790	7	Eastern
New Hampshire	1,371,246	3	Eastern

Fig. 3 The ten least populated states and the number of ransomware attacks since 2013



Fig. 4 Ransomware data among small and medium sized businesses [5]

important organizations present in these areas, or another factor? To answer these questions, we used the second map (Fig. 4).

The map we used was found in an article on the StateScoop website. It allows the user to choose a state and view when and where the attack happened and a news article on that specific attack. We went through each state and recorded the

number of attacks seen in that states since 2013 and divided them into the eastern and western states. According to this information, the western states have had 139 ransomware attacks since 2013, while the eastern states have had 212 [3]. We then found the population of each side; this showed that the western states have a population of over 142,000,000 and the eastern states have a population of over 188,000,000 [4] (Fig. 5).

Fig. 5 The run-times of the CovidLock simulation code and average in seconds

```

1  3.823524236679077
2  2.4443843364715576
3  3.436765193939209
4  4.259863376617432
5  3.046236038208008
6  3.1585659980773926
7  3.0223121643066406
8  3.440958023071289
9  3.5787339210510254
10 3.4430699348449707
11 3.463367462158203
12 3.626089572906494
13 3.5173537731170654
14 3.323716163635254
15 3.53839111328125
16 3.179046154022217
17 3.8160619735717773
18 3.900728940963745
19 3.081648588180542
20 3.258338689804077
21 2.9954795837402344
22 3.589646100997925
23 3.900679111480713
24 4.125925302505493
25 3.7397820949554443
26 3.667924832081389
27 2.897923332081387
28 4.013785937402935
29 3.836640927462845
30 4.10095362663269
31 2.684504747390747
32 3.721098232269287
33 2.879676580429077
34 2.6829681396484375
35 3.329174041748047
36
37 AVERAGE: 3.4435805213067 seconds

```

After viewing this information, we noticed that the two most populated states, California and Texas, experienced the most ransomware attacks during this time period. However, we realized that population did not always seem to be a big factor in the amount of attacks; some states with a higher population experienced less attacks, while some states with a lower population experienced more attacks. Another key piece of information found was that the eastern states are more populated than the western states. This explains why many attacks have occurred in the eastern states (Fig. 6).

2.2 Reaction Time

One of the most common ways ransomware infects a user’s device is through the user’s own actions. The user is most likely unaware they’re infecting their device, but without their input, most ransomware would fall flat on its face. Those savvy enough with technology can spot ransomware when they notice strange requests or links that lead to a site they don’t recognize (Fig. 7).

However, to spot these inaccuracies requires a keen eye and prior knowledge. The common users often lack these qualities and are more prone to believing what they see or read. In fact, a survey run by Deloitte discovered that “of 2,000 consumers in the U.S.,” about 91 percent accept “legal terms and services conditions without reading them” [6]. For these users, it will be especially difficult to notice ransomware until it is too late (Fig. 8).

To help these users, the CovidLock simulation code was timed. From the moment the simulation begins, a timer runs until the lock screen demanding a ransom in

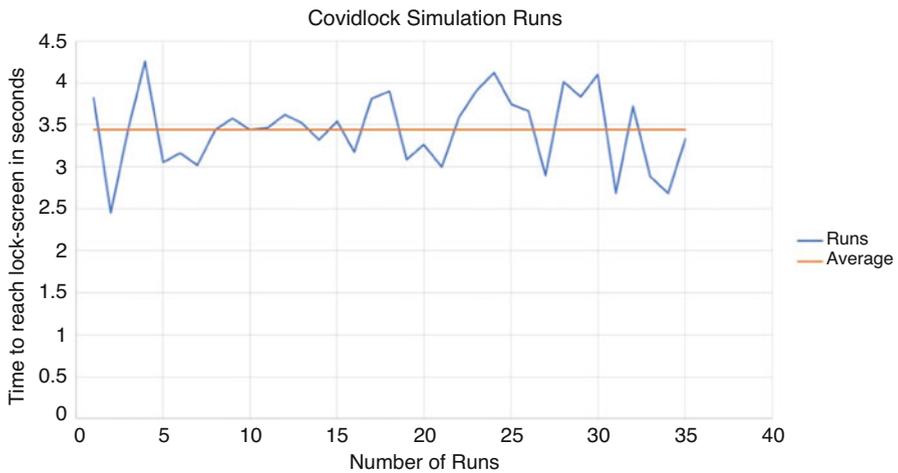


Fig. 6 Graph of the run-times and average listed in Fig. 5

```

print("--Covidlock Mitigation--")
print()
print()

# Note: The keys are the names of the apps, while the values are the companies who made them
# The idea is that a program can check the publisher of an app to determine whether they are trustworthy or not
Whitelist = {"TikTok": "ByteDance", "Citymapper": "Citymapper Limited", "Trello": "Atlassian", "Google Podcasts": "Google LLC",}
Blacklist = {"Coronavirus Tracker": "https://coronavirusapp[.]site/mobile.html"}

# You are an Android user, looking to download some apps from the app store.
# Through this mitigation program, the user can check whether the app is from a verified/trusted source or not

# User wants to download TikTok
print("Is TikTok from a trusted source?")
if "TikTok" in Whitelist.keys():
    print("TRUSTED SOURCE. Download.")
elif "TikTok" in Blacklist.keys():
    print("UNTRUSTED SOURCE. Do not download.")

print()

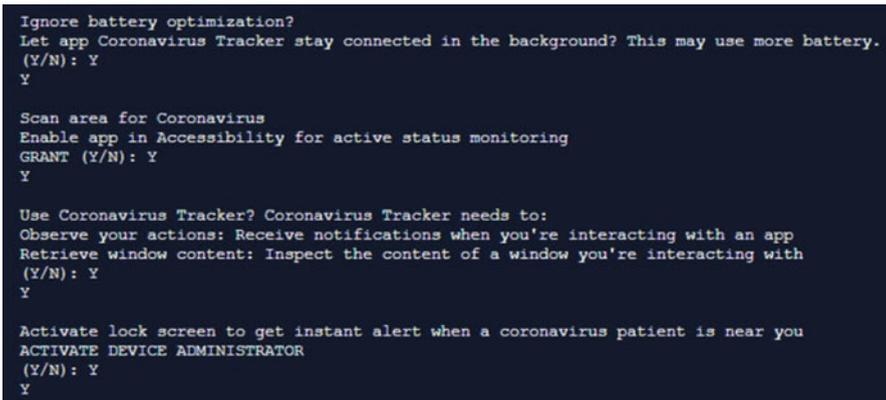
# User has downloaded Citymapper, and has opened the app for the first time
print("Is Citymapper from a trusted source?")
if "Citymapper" in Whitelist.keys():
    print("TRUSTED SOURCE.")
elif "Citymapper" in Blacklist.keys():
    print("UNTRUSTED SOURCE. Deleting application now.")

print()

# User wants to download Coronavirus Tracker
print("Is Coronavirus Tracker from a trusted source?")
if "Coronavirus Tracker" in Whitelist.keys():
    print("TRUSTED SOURCE. Download.")
elif "Coronavirus Tracker" in Blacklist.keys():
    print("UNTRUSTED SOURCE. Do not download.")

```

Fig. 7 A simple mitigation code to combat ransomware



```

Ignore battery optimization?
Let app Coronavirus Tracker stay connected in the background? This may use more battery.
(Y/N): Y
Y

Scan area for Coronavirus
Enable app in Accessibility for active status monitoring
GRANT (Y/N): Y
Y

Use Coronavirus Tracker? Coronavirus Tracker needs to:
Observe your actions: Receive notifications when you're interacting with an app
Retrieve window content: Inspect the content of a window you're interacting with
(Y/N): Y
Y

Activate lock screen to get instant alert when a coronavirus patient is near you
ACTIVATE DEVICE ADMINISTRATOR
(Y/N): Y
Y

```

Fig. 8 A simulation of the requests the ransomware makes

cryptocurrency is reached. Thirty-five manual runs were conducted, each trying to reach the lock screen by agreeing to every request the CovidLock application asked as fast as possible. This was done to acquire human reaction times akin to the 91 percent of users who accept every request blindly just to access the application (Fig. 9).

```
--PHONE LOCKED SCREEN--
YOUR PHONE IS ENCRYPTED: YOU HAVE 48 HOURS TO PAY 100$ in BITCOIN OR EVERYTHING WILL BE ERASED
1. What will be deleted? your contacts, your pictures and videos, all social media accounts will be leaked publicly and the phone memory will be completely erased
2. How to save it? you need a decryption code that will disarm the app and unlock your data back as it was before
3. How to get the decryption code? you need to send the 100$ in bitcoin to the address below, click the button below to see the code
NOTE: YOUR GPS IS WATCHED AND YOUR LOCATION IS KNOWN, IF YOU TRY ANYTHING STUPID YOUR PHONE WILL BE AUTOMATICALLY ERASED
Enter Decryption Code: █
```

Fig. 9 A simulation of the CovidLock ransom lock screen

Most of these times ranged between 3 and 4 s, with the fastest time being around 2.444 s and the longest time around 4.259 s. The average of these 35 run-times, which serves as a decent ideal reaction time for the mitigation code, was around 3.443 s long. This is a reasonable time frame for a mitigation to react and save users from falling victim to ransomware.

3 Overview of the Simulation

There are two primary roles in this simulation, the role of the target and the role of the attacker. In this simulation, the target is an Android user that is frightened over the COVID-19 pandemic. To see how badly their area has been hit by the pandemic, the user has downloaded the attacker app from the Android app store, Coronavirus Tracker. This simulation presumes that the user's Android device will not detect the ransomware installed within Coronavirus Tracker. The simulation goes through each step of the CovidLock process word-by-word, ending with the lock screen asking the user for a ransom.

To help fend off this sort of ransomware, we've developed a simple code for a hypothetical application to aid users. The app has a directory of whitelisted and blacklisted applications on the Android app store. Apps that come from trusted publishers and sources, such as Google, would be among the whitelisted directory. Apps that come from unverified sources, like the CovidLock application, would fall into the blacklisted directory.

This mitigation application would have the ability to check apps before the user downloads them to see if they are safe to install. If the user has already downloaded an application or somehow missed the option to check whether the application is from a trusted source, the mitigation app can check the new application when it is first opened. In this case scenario, it is imperative that the mitigation application notices any blacklisted applications within the calculated reaction time stated above. Otherwise, in the instance of CovidLock, the user may speed through and accept the ransomware's requests and be locked out of their device before the mitigation app can stop it.

Not every user has the technical savvy to recognize suspicious behavior or requests from applications. For those users, an application that handles the detection and removal of ransomware is vital to protect the data and memory stored on their devices. To test this mitigation app's effectiveness in a real-life scenario, it'd be ideal to test it in states more prone to ransomware attacks. There'd be greater chances to pit the mitigation app against a variety of ransomware attacks and allow the mitigation application to fill its whitelist and blacklist directories.

4 Simulation Methodology

Ransomware comes in many different forms, and we've focused on lock screen attacks like CovidLock due to its topical nature. Our aim was to offer a potential hypothetical solution that'd lessen the stress for users who aren't as knowledgeable on malware like ransomware. Especially amidst a global pandemic, it's expected that users may act out of fear and desperation to try and protect themselves. This of course puts them at risk of cyberattacks like ransomware, which prey on ignorant users.

As a recent form of ransomware, there was initially only a handful of data available about the CovidLock ransomware. The new influx of information, such as Desai's "technical walkthrough of the app" allowed for a proper, point-by-point simulation of the CovidLock ransomware to be created through Python [7]. This allowed for a safe simulation of what the ransomware did and allowed for reaction times to be gauged.

A simple mitigation code was created as a potential solution to ransomware like CovidLock. Since the mitigation code required a test of a user's potential fastest run-time, using SimPy, a simulation framework for Python, was not possible. SimPy operates on fixed time intervals, meaning that recording individual unique run-times would not be possible. These unique run-times, simulating a user reaching the lock screen as quickly as possible, were necessary to give a time frame for how quickly a mitigation would need to act.

To see the spread of ransomware attacks in the United States, research was conducted on which states suffered a greater number of ransomware attacks. This information would provide potential testing grounds for the mitigation code in the form of states with more cases of ransomware attacks.

5 Comparison to Other Mitigation Methods

Ransomware has been a problem in the world for a long time. So, there are already ways to prevent or mitigate these problems. However, because of the constant changes and enhancements to this technology, these methods do not always work with newer ransomware.

In a paper written on the history and mitigation of ransomware, the authors explain a few ideas that may decrease the impact of ransomware. These include backing up information on your computer, avoiding suspicious attachments and links in suspicious emails, keeping the system patched and up-to-date, and unplug the system when the user first identifies that it is infected. Other methods mentioned include understanding the danger of malware, creating policies that prevent malicious code from entering the network, and having requirements like mandatory password changes and annual security awareness training [8]. Each of these methods is great ways to prepare for a ransomware attack. However, user error occurs, and these security measures will not always stop attackers from gaining access into the network.

In another paper, the author explains the use of firewalls and security software to prevent these attacks [9]. While firewalls are a great way to keep malware out of the network, they will not be able to stop the malware when it has already passed through it. This is different in our mitigation method. With our new method, we can continue to monitor the app when it has been downloaded. This will improve the protection of the user's computer and the information on it. Security software, such as antivirus, are a good way to detect malware. However, attackers have become more aware of the restrictions these have and are able to navigate through some of the security software that is popular among users. With a new way to stop malware attacks, these invasions will be slowed because the attackers will need a new way of entry.

6 Attack Example

Like most forms of ransomware, CovidLock relies on the user's input to implement its malicious code. A Python code was created to simulate this specific attack, for the purpose of study for this report and to develop a proper mitigation for this brand of ransomware. As Goud [10] explains, this ransomware pretends to help users "track down" those infected with COVID-19 "once they are in their vicinity using heatmap visuals." To supposedly accomplish this, the infected application asks users for certain privileges to embed itself into their Android devices. These include asking for "battery optimization" as well as Android's "accessibility feature" and most notably "administrator privileges" [7].

To users unaware of the technicalities or those simply in a rush to scan the area for those infected with the virus, the requests seem harmless. A request for administrator privileges, however, is a red flag for some users who understand what the application is asking. Agreeing to each request is necessary for the ransomware to offer users a chance to scan the area. However, clicking this will only activate the ransomware and leave their Android device stuck on the lock screen. This is a point of no return for most users, as without a proper decryption code, they are left with no choice but to pay the ransom in hopes their device will be unlocked.

The mitigation code explained in the prior sections would theoretically stop users from ever reaching this point of no return. This is the case because the ransomware triggers once a user uses the app's COVID-19 scanner. Ideally, the mitigation code would stop users from even granting the CovidLock app any permissions by preventing them from downloading the application at all. As Desai notes, the source for this ransomware is a site with the URL "hxxp://coronavirusapp[.]site/mobile.html," which is not a known or trusted source [7]. As such, the mitigation would have the COVID-19 Tracker in its blacklisted directory and notify users prior to download or upon opening the application of its unverifiability and prompt for the ransomware's deletion.

7 Conclusion

Ransomware is a growing problem in the world today. With new types of malware arising, new, more sophisticated solutions must be created. Knowing the risks of this malicious code and by using firewalls and security software will not keep users safe for very long. By implementing this different way of mitigation and adding to it, we will not only be able to prevent the problem of CovidLock, but we will also be able to prevent other malware from causing too much harm to users.

References

1. T. Saleh, CovidLock update: Deeper analysis of coronavirus android ransomware. Domain-Tools, 16 Mar 2020. www.domaintools.com/resources/blog/covidlock-update-coronavirus-ransomware
2. The Ransomware War by PC Matic – Google My Maps. www.google.com/mymaps/viewer?mid=1UE6Nko9iRG1tLci_AeqqsxxzGzs
3. B. Freed, Ransomware attacks map chronicles a growing threat. StateScoop, 24 Apr 2020, statescoop.com/ransomware-attacks-map-state-local-government/
4. US States – Ranked by Population 2020, 24 Apr 2020. worldpopulationreview.com/states/
5. UpGuard. 17 Ransomware Examples. UpGuard, UpGuard, 9 Dec 2019. www.upguard.com/blog/ransomware-examples
6. C. Cakebread, You're Not Alone, No One Reads Terms of Service Agreements. Business Insider, Business Insider, 15 Nov 2017. www.businessinsider.com/deloitte-study-91-percent-agree-terms-of-service-without-reading-2017-11
7. S. Desai, CovidLock: Android Ransomware Walkthrough and Unlocking Routine. Zscaler, 16 Mar 2020. www.zscaler.com/blogs/research/covidlock-android-ransomware-walkthrough-and-unlocking-routine
8. R. Richardson, M. Max, North. Ransomware: Evolution, mitigation and prevention. *Int. Manag. Rev.* **13**(1), 10 (2017)
9. M.A.H. Saeed, Malware in computer systems: Problems and solutions. *Int. J. Inf. Devel.* **9**(1), 1–8 (2020)
10. N. Goud et al., Details of CovidLock Ransomware and Czech Hospital Infection. *Cybersecurity Insiders*, 16 Mar 2020. www.cybersecurity-insiders.com/details-of-covidlock-ransomware-and-czech-hospital-infection/

The New Office Threat: A Simulation of Watering Hole Cyberattacks



Braeden Bowen, Jeremy Eraybar, Iyanuoluwa Odebode, Douglas D. Hodson, and Michael R. Grimaila

1 Introduction

At the current speed of our global propagation of network computer systems and capabilities, public and private, there is a constant stream of new technological threats. Malicious cyberattacks have become a norm of the information age, disrupting the daily functions of governments, national emergency and defense systems, the global economy, and common daily functions.

In order to support a growing reliance on technology, many companies and global organizations all shift day-to-day operations to cloud-based services, creating many vulnerabilities known and unknown. In addition, the increase of bring your own device (BYOD) policies at the institutional levels has increased the threat levels everywhere. Just for educational institutions by themselves, the policies and clearance of students and staff on their capabilities have increased drastically. Educational institutions now allow students and staff to access the institution's network and perform various tasks through their data bases. An astounding percentage of the usage of personal devices in the educational environment from various countries was found in the 2014 Educare report [2, 6]. It was reported that the usage of personal devices in the educational environment is a must have, where 90%

B. Bowen (✉) · J. Eraybar
Wittenberg University, Springfield, OH, USA
e-mail: bowenb3@wittenberg.edu; eraybarj@wittenberg.edu

I. Odebode
Oak Ridge Institute for Science & Education, Wright Patterson AFB, Dayton, OH, USA
e-mail: iyanuoluwa.odebode@afit.edu

D. D. Hodson · M. R. Grimaila
Air Force Institute of Technology, Wright Patterson AFB, Dayton, OH, USA
e-mail: douglas.hodson@afit.edu; michael.grimaila@afit.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_4

of just undergraduates own a laptop and 86% own a smartphone [3]. Thus, it's extremely important for companies, institutions, and global organizations alike to be knowledgeable about the ever-growing vulnerabilities of their systems [7].

With most polices extending to BYOD, the threat of an advanced persistent threat (APT) has never been more prevalent. APT combines both social engineering and information gathering in order to have the best success rate for the hacker. Targeting businesses, offices, and educational organizations, cyberattackers employ a style of attack called a "watering hole" attack [8, 9]. A watering hole is a computer hacking strategy which was identified as recently as 2012 by the RSA security in the VOLTO campaign. Current research shows that watering hole attacks are rapidly snowballing [3]. The goal of a watering hole attack is not to expose individual systems to malicious malware: instead, attackers exploit known, commonly used, and trusted sites catered to their victims [3]. Hackers that use this technique observe websites used by a target group or organization and inject harmful malware into the site. With this attack, a single visitor could jeopardize an entire company, organization, region, industry, or institution. The attack fundamentally mirrors its namesake, a dominant predator waiting near a water source to pounce on their prey. Watering hole attacks can affect anyone, which is why they can be so discrete and successful.

Organizations that practice BYOB may especially attract APT attacks due to the users' lack awareness to such attacks [3]. Consequently, it is difficult for security experts to produce adequate techniques of preventing these watering hole attacks.

Hackers create or use common Internet tracking tools such as "add this and kiss metrics" to identify sites that targets visit recurrently [4]. The predictability of target behavior allows attackers to develop a recurring process for attack.

Hackers start by injecting code into the selected sites, which then drop malware on vulnerable systems when the "watering hole" is accessed by a target user [4]. Next, they use the dropped malware to initiate the malware "activities," relaying information from the compromised system back to the attacker. With this information, the cycle of information gathering from the initial is repeated, expanding the attacker's understanding of the attack surface. With additional information, hackers inject malware in a plethora of specific sites and wait for the target users to visit. Once the target is identified to have visited the site, hackers identify the vulnerabilities such as outdated antiviruses or systems. Then, leveraging the "drive-by download technique," the hacker doesn't need the user to click on or download any files [4]. A small amount of code is downloaded discreetly in the background based upon the user's access rights. This access is the final step of the attack process: to gain access to confidential information. Hackers then take the intellectual property as their own to gain devastating information from the original target [4].

Since its conception in 1976, DEVS has become a prevalent model for creating complex and dynamic systems using discrete-based levels of constructs [5, 11]. The DEVS model creates a timed sequence of events that are implemented into its system, thus creating rapid fluctuations to the state of the desired system [10, 13]. These events can be created externally or internally. Consequently, every state of the system is based upon the proceedings of the last event. Between these events,

the state does not change which create a constant state of progression. Thus, the DEVS system allows the skipping of intermediate time compared to the alternative of discrete models [1]. The main advantage of DEVS is the ability to implement reproductions inside of its system components, theoretically creating an advantage in the system of modeling.

2 Methodology

With the principles of the DEVS model in mind [12, 14], we developed our own simulation environment that is dedicated to managing three factors of a watering hole attack:

- (1) Number of participants
- (2) Number of shared websites
- (3) Length of simulation

The watering hole simulation model, written in Python, simulated a virtual work environment. In order to increase the attack surface and more effectively simulate random human interactions with objectively linear digital processes, random numbers were selected at the start of each simulation event to choose the elements that would be involved in the attack for the duration of the event.

The simulation was comprised of a website list containing Website objects, which contained a “File” object list, as well as an “Office” list containing Office Worker objects which themselves contained a list of “File” objects. A simulation object managed a Python Threading Timer object with a predetermined number of seconds derived from the main method. The Simulation object generated a random event for a randomly chosen Office Worker every .1 s and then slept for .2 s, for an event length of .3 s.

File objects contained only cursory information and were used for tracking movement of infection across the simulation rather than holding data. The objects stored their own name, the Office Worker or Website object that created it, and a Boolean variable that determined whether or not the object was “infected.” If the object (Website or Office Worker) that generated the File object was infected, the File object became infected, too.

Website objects had marginally more functionality, with the ability to receive, give, and create new files. Of the list generated by the Simulation object at the start of each trial, only one of the predetermined numbers of websites in the list was infected: this site served as the entry point for the watering hole attack, while the remaining number of websites and total sum of Office Workers served as the attack surface. If a Website object received an infected File, the website would from then on be infected, as well. The actions and movement of the Office Worker objects inside the “Office” list were the primary focus of the simulation. Each was randomly assigned an IP address by the Simulation object before the simulation timer began.

During the simulation, each Office Worker object had five potential actions for each event cycle: “use site,” which had a 50% chance of infecting the chosen worker if an infected website was randomly selected; “upload file,” which added a random File object from the chosen Office Worker’s File list to the chosen Website object’s file list; “download file,” which copied a random file from a randomly chosen Website; “share file,” which copied a File directly between two Office Workers; and “virus scan,” which had a 25% chance to “disinfect” an Office Worker and a 50% chance to disinfect each of its File objects individually.

For each event cycle, a log was produced marking the time, Office Worker, action, and target. Whether or not any object was infected during an event was managed by the simulation, but not printed until after the Python Threading Timer was finished, where relevant metadata was formatted and outputted by the Simulation object via the main method. Infection rate was calculated by dividing the number of infected Office Workers in the simulation by the total number in the office list.

One hundred workers and three websites, one of which was infected by default, were chosen as the base variables for the simulation. The base time variable, 48 s, was chosen as a simplification of an 8-hour workday, consisting of approximately 480 min. Each Simulation object event (every .1 seconds + .2 second rest) represented 3 min of a workday, when a single Office Worker event occurred.

Variations from each base variable were tested for their relative ability to impact the infection rate determined by the simulation. Office Worker counts of 25, 50, 100, 150, and 175; Website object counts of 1, 2, 3, 4, and 5; and durations of 12, 24, 48, 96, and 192 were each tested ten times and averaged. Each average was graphed.

2.1 Results

The base variable results for the Office Worker variation simulations produced an average infection rate (AIR) of 41.9% over a 480-minute (48 s) workday, with a distribution of 50% to 27%. For 25 Office Workers under the same conditions, the AIR was 77.2%, with a distribution of 88% to 60%. Fifty Office Workers had an AIR of 62.99%, with a distribution of 78% to 54%. One hundred fifty Office Workers held an AIR of 31.2%, with a distribution of 35.3% to 28%. The largest tested scale of Office Workers, 175, had an AIR of just 25.5%, with a distribution of 34.3% to 17.7%. The results suggest a linear relationship between higher numbers of Office Workers and lower numbers of infection over a fixed period (Fig. 1).

The base variable results for the Website variation simulations produced an average infection rate (AIR) of 41.2% over a 480-minute (48 s) workday, a 0.7% deviation from the average results with the same variables from the Office Worker variation simulations. This suggests that because of the volatile nature of the behavior chosen by the simulation, the results could only be reasonably deemed to be similar and repeatable. One Website object returned an AIR of 51.7% over the same time period, with a distribution of 56.9% to 46%. 2 Websites produced an AIR of 43.6% with a distribution of 49% to 34%. 4 Websites had an AIR of

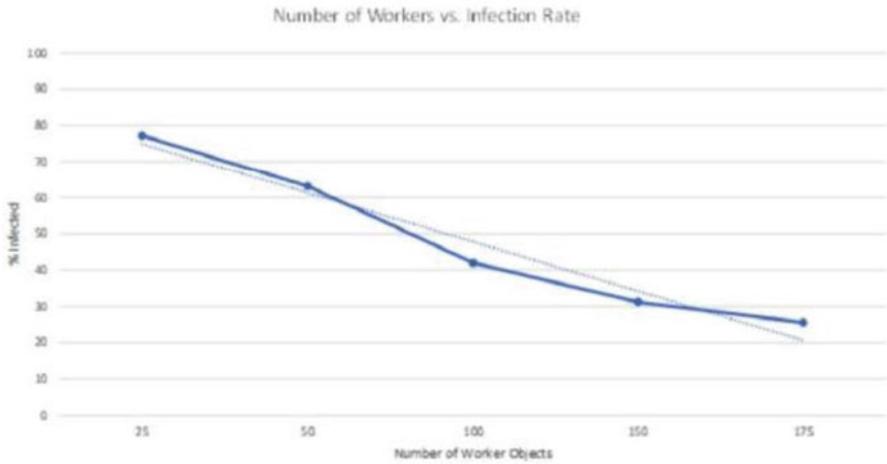


Fig. 1 Office Worker count variation

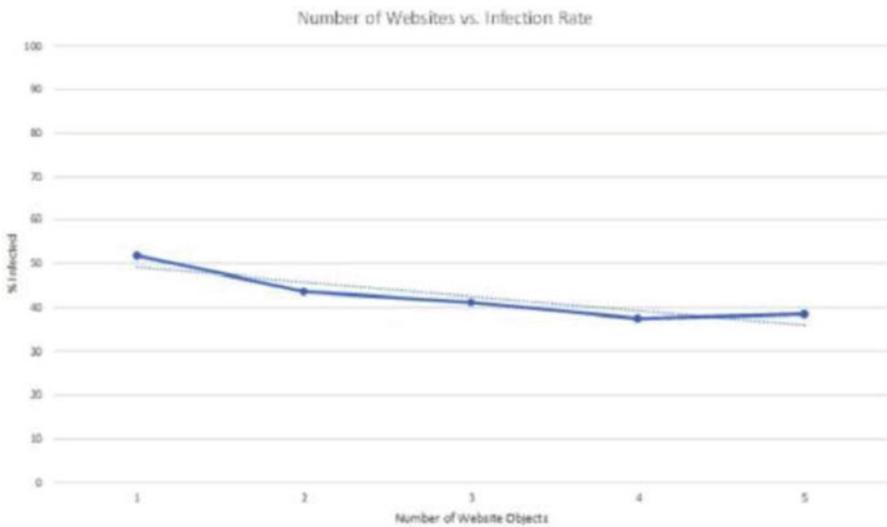


Fig. 2 Website Count Variation

37.5%, with a distribution of 44% to 32%. The original set of data for a simulation of five Websites produced unexpectedly high result of a 41% AIR, so the simulation was repeated to eliminate potential extreme outliers. Unexpectedly, the simulation produced an AIR of 38.5% for five Websites (with a distribution of 41% to 31%), a full 1% higher than the AIR of a four Website simulation (Fig. 2).

Although the results were not included in the data, a simulation with six Websites had an AIR of 38.1% with a distribution of 49% to 33%, suggesting that after a

certain point, an increase in the attack surface of a watering hole attack does not significantly impact the AIR.

Duration Variation (Fig. 3). The base variable results for the duration variation simulations produced an average infection rate (AIR) of 40.3% over a 480-minute (48 s) workday, 1.6% lower than the first base AIR and 0.9% lower than the second, corroborating the earlier suggestion of volatile, random behavior. Starting from a time of 12 s, a quarter of normal 8-hour workday, the AIR was 12.4%, with a distribution of 18% to 5%. In durations of 24 s, half of a workday, the AIR was 24.98% with a distribution of 30% to 18%. Durations of 60 s, a day and a quarter, produced an AIR of 48.2% with a distribution of 55% to 40%. For durations of 72 s, a day and a half, an AIR of 54.4% was produced, with a distribution of 61% to 50%.

2.2 Analysis

The results of the simulation largely behaved as expected. In general, the results suggested that the more machines there are in an office environment, the fewer devices will become infected in a standard amount of time. In this environment, the model suggests diminishing returns on increasing numbers of machines in an office environment: 25-worker to 50-worker simulations showed an AIR diminishing of 14.2%, while later simulations of 15-worker to 175-worker, the same 25-worker differential, showed an AIR decrease of just 5.7%, 2.5 times less than the first increment between simulations. These results could mean that after a certain point, an increase in the number of workers would have an overall negligible effect

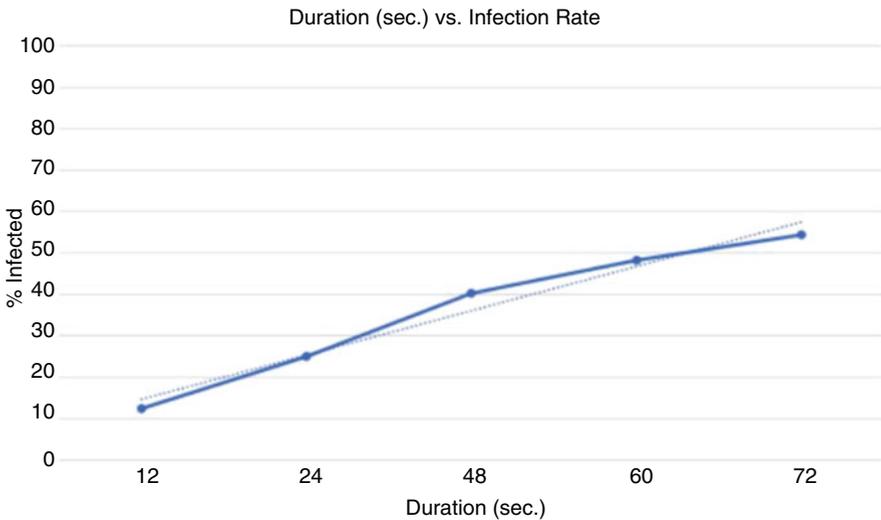


Fig. 3 Website count variation

on the infection rate. The results for the website variation behaved similarly and produced diminishing returns more rapidly: after simulations with 33% of the websites starting out infected, which produced an AIR of 41.2%, results consistently tended to hover below or at 40% AIR, even for simulations with as low as a 17% of starting sites being “watering holes.” The simulation was designed to measure AIR within a boundary of time. If left unchecked for an indefinite amount of time, though, the infection would easily overwhelm an organizational structure. As unchecked time continues, the simulation again suggests diminishing returns for newly infected devices, as already-infected devices continue to function normally within the simulation. From 12-second to 24-second simulations, the AIR grew by 12.58%. The 60-second to 72-second simulations, though, the same 12-second time differential, saw an AIR increase of only 6.2%, less than half the growth between the shorter simulations. Still, given the tested conditions of the simulation, the model suggests that the entirety of a 100-person office could be infected in under 5 workdays or 240 s.

3 Conclusion

With the continuing rise in commonality of advanced persistent threat (APT) attacks like the watering hole attack, it is important to study the behavior and repercussions of this attack style. This simulation offers some insight into the speed at which malware may be transmitted between members of a target group and investigates variables that may bolster or hinder the spread of the malware through the system. Our analysis of the results suggests that time and concentration of the target group are the most important factors for spread from a single watering hole source but that for any given variable the returns are diminishing as the size of a target, number of commonly used sites, and allotted time increase.

3.1 Future Research

With the continuing rise in commonality of advanced persistent threat (APT) attacks like the watering hole attack, it is important to study the behavior and repercussions of this attack style. This simulation offers some insight into the speed at which malware may be transmitted between members of a target group and investigates variables that may bolster or hinder the spread of the malware through the system. Our analysis of the results suggests that time and concentration of the target group are the most important factors for spread from a single watering hole source but that for any given variable the returns are diminishing as the size of a target, number of commonly used sites, and allotted time increase.

More research should be done on the spread of viral, watering hole-style attacks. This simulation environment offers a wealth of data for later analysis—algorithms

could be written to retrace log data produced by the simulation to determine infection sources and methods of transmission. The existing simulation could also be rewritten to run until all members of the “Office” list are infected. Additionally, less time between events could alter the results of the simulation and should be tested. Although the simulation environment worked to emulate the randomness of human interaction with static processes, human timing is not as static in a true work environment as the simulation presumes. Randomly generated event cycle lengths should also be considered in future research. Additional research should be done on the level of diminishing returns in watering hole attacks. Is there a point at which an organization or department’s website is not worth targeting at all? What conditions would produce the optimal infection rate for an attacker to compromise as much data as possible?

References

1. G. Gordon, A general purpose systems simulator. *IBM Syst. J.* **1**(1), 18–32 (1962)
2. R. Grimes, Watch out for watering hole attacks—hackers’ latest stealth weapon, 2013. [Online]. Available: <https://www.csoonline.com/article/2614643>. Accessed 15 June 2021
3. K.A. Ismail, M.M. Singh, N. Mustafa, P. Keikhosrokiani, Z. Zulkefli, Security strategies for hindering watering hole cyber crime attack. *Proc. Comput. Sci.* **124**, 656–663 (2017)
4. N. Krithika, A study on wha (watering hole attack)—the most dangerous threat to the organisation. *Int. J. Innov. Sci. Eng. Res.* **4**(8), 196–198 (2017)
5. O. Kupreev, E. Badovskaya, A. Gutnikov, Ddos attacks in q1 2019 (2019)
6. M. Mimoso, Council on foreign relations website hit by watering hole attack, ie zero-day exploit, December 2012. [Online]. Available: <https://threatpost.com/council-foreign-relations-website-hit-watering-hole-attack-ie-zero-day-exploit-122912/77352/>. Accessed 15 June 2021
7. R. Olivarez, Watering hole attacks: Tips on outsmarting the hackers. May 6, 2014. [Online]. Available: <https://wp.nyu.edu/connect/2014/05/06/watering-hole-attacks/>. Accessed 15 June 2021
8. M. Bacon M. Rouse, Watering hole attack; Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Watering_hole_attack. Accessed 15 June 2021
9. G.V. Umoh, N.P. Paul, The adverse effect of watering hole attack in distributed systems and the preventive measures. *International Journal of Computer Trends and Technology (IJCTT)* **23**(4), 162–165 (2015)
10. Y. Van Tendeloo, H. Vangheluwe, An overview of pythonpdevs. *JDF 2016-Les Journées DEVS Francophones-Théorie et Applications*, pp. 59–66 (2016)
11. Y. Van Tendeloo, H. Vangheluwe, An evaluation of devs simulation tools. *Simulation* **93**(2), 103–121 (2017)
12. B.P. Zeigler, A. Muzy, From discrete event simulation to discrete event specified systems (devs). *IFAC-PapersOnLine* **50**(1), 3039–3044 (2017)
13. B.P. Zeigler, J.J. Nutaro, Towards a framework for more robust validation and verification of simulation models for systems of systems. *J. Def. Model. Simul.* **13**(1), 3–16 (2016)
14. B.P. Zeigler, T.G. Kim, H. Praehofer, *Theory of Modeling and Simulation* (Academic Press, 2000)

Simulation of SYN Flood Attack and Counter-Attack Methods Using Average Connection Times



Hai Vo, Raymond Kozlowski, Iyanuoluwa Odebode, Douglas D. Hodson, and Michael R. Grimaila

1 Introduction

With the growth and increased usage of our Internet resources during this time, prevention and detection of cybersecurity issues or attacks have become one of the utmost important aspects. Due to the nature of increased usage, one issue that can be leveraged along with it is denial-of-service attacks. While the heavy load of data transmission can certainly affect the server, a distributed denial-of-service (DDoS) attack is a way to try to make online services unavailable to other users by overwhelming it with traffic from multiple sources. This kind of attack can target various kinds of services, from financial online systems to entertainments, preventing people to gain access to valuable information.

There are multiple reasons why this attack is carried out, from too competitive gamers trying to gain an advantage among other gamers [3, 4, 9], to be used as cyberwarfare in politics. Some of the attacks can include UDP flood, HTTP flood, ping of death, etc. In UDP flood attacks, attackers can generate fake traffic that can consume the bandwidth and overwhelm the system. Since the UDP protocol does not utilize handshaking techniques, this attack can easily be carried out with huge amount of traffic [7, 9–11] (Fig. 1).

H. Vo · R. Kozlowski
Wittenberg University, Springfield, OH, USA
e-mail: bowenb3@wittenberg.edu; eraybarj@wittenberg.edu

I. Odebode
Oak Ridge Institute for Science & Education, Wright Patterson AFB, Dayton, OH, USA
e-mail: iyanuoluwa.odebode@afit.edu

D. D. Hodson (✉) · M. R. Grimaila
Air Force Institute of Technology, Wright Patterson AFB, Dayton, OH, USA
e-mail: douglas.hodson@afit.edu; michael.grimaila@afit.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_5

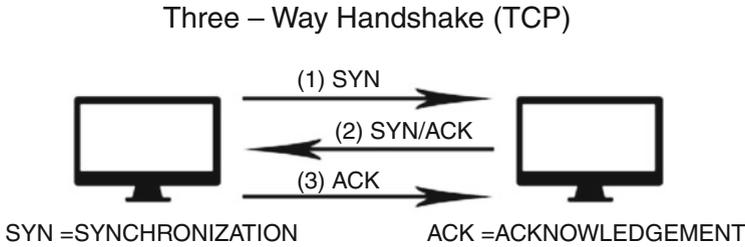


Fig. 1 Three way handshake in tcp protocol

In our case, we will be looking at SYN flood attacks, which target the TCP protocol. The TCP protocol is a guaranteed delivery protocol because it provides flow control and reliable data delivery with acknowledgments between both the server and the client. Specifically, a SYN flood attack relies on leveraging the way TCP communication. Usually, a client sends a SYN packet to the server to ask for a TCP connection. The server would then send back a SYN-ACK packet to the client to acknowledge the client connection. The client then sends to the server an ACK packet, acknowledging that prior package, and establishes the connection between the two. The process described above is usually known as a “three-way handshake” [3, 13].

In the three-way handshake, when the server waits for the ACK packet from the client to establish the connection, it is called the half-open state. In this state, the server keeps waiting for an ACK packet back from the client to finish the handshake. This is managed with a backlog queue, which can time out the connections that stay open for too long. However, when this queue is too full of request, in addition to the memory constraints of the server, this can lead to denial-of-server on the server end. This means new connections cannot be made and instead of doing the three-hand shake, the server cannot respond to any SYN packet from real user who wants to connect. What makes this attack more popular is that the attack packets are the exact same SYN packet used in a legitimate connection, making it very hard to recognize the attack. In fact, SYN flood attack is still one of the most popular attacks, accounting for 92.57% of attacks in Q1 of 2020 [3, 8] (Fig. 2).

2 Methodology

We implemented this experiment using the DEVS framework [12, 14].

DEVS helps to define the behavior of systems; we are able to define transitions between different sequences and also define how this system respond to the external events and also how these entire system is evaluated [5, 6].

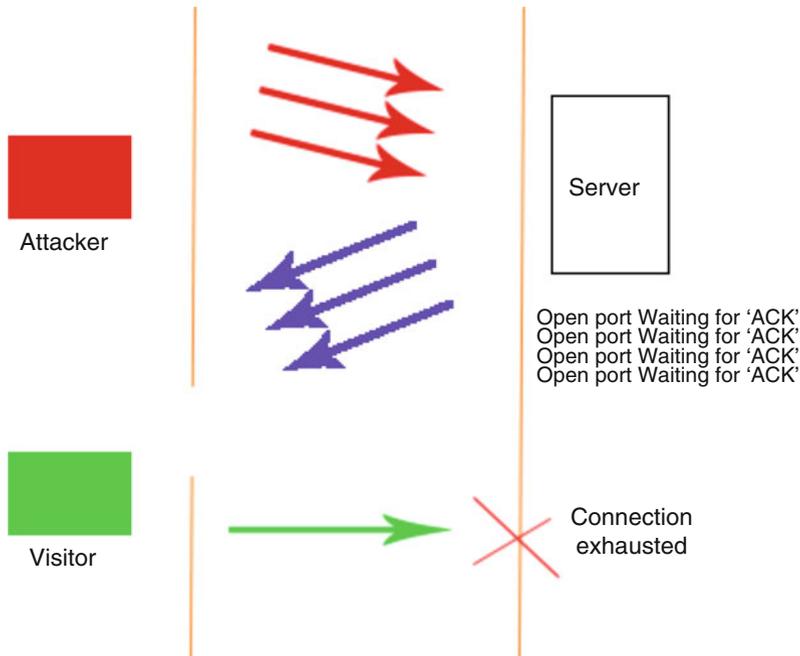


Fig. 2 TCP SYN flood attack

DEVS has been popularly used to simulate the generation and the storage of biogas that is produced through organic waste. The waste is further used to produce electrical energy in a supply chain environment and also to validate multiple scenarios as it affects the generation and consumption of this biogas [1].

Our working SimPy environment is first set up with two classes, client and server. For my client class, a random IP is generated, with one SYN packet set to be sent to the server for the initial handshake. There is also an `attacker_value`, which then generates between 2 and 7 SYN packets to send to the server.

For my server class, the server is set up with a server as a finite resource of 100-packet size, as well as lists containing the time for successful handshake connections and half-open connections to clients. After setting up the classes, I create a handshake method, which takes a server and a client object as the arguments. This method proceeds to send an SYN request to the server from the client. If the client's `attacker_value` is set to false, the `TCP_server` then release the request from the queue, finishing the handshake and start the transmission of the files, which take between 1 and 6 s. Else if the `attacker_value` is set, then the client object will request and send multiple SYN packets, in an attempt to fill up the queue.

There have been several methods implemented to counter these kinds of attacks, including using an SYN cache [9], IP table firewalls [10], or “SYN cookies shake anomalies” written by Martine Bellaïche and Jean-Charles Grégoire [2, 9, 10]. For

this stimulation, I am collecting data in three different ways, freeing up some of the processing queue using two techniques, and not doing so for the other one. Both counter-attack methods are based on the paper “SYN flooding attack detection by TCP handshake anomalies” written by Martine Bellaïche and Jean-Charles Grégoire [2, 7, 9, 10].

The first method of gathering data is to not apply any time-out for the request and not free up anything, in order to get a baseline of how it would affect the system. The second method is to append all the connection times, measured between the reception of the SYN packet from the client to the end of the transmission between the server and the client, as well as any connection times of unresolved packets still stuck in the queue. One issue to this method is that each SYN packet received is set to take 0.5 s to complete due to technical difficulties, so this might not provide as much accurate data as it should be. However, it should still be a good measure. The third method is to append only successful handshake connection times. This would provide a closer to real-life server processing time. However, it does ignore the pending SYN request times from the attackers. Using three different ways to set up the time-out for the queue, I create the server_time_out method, which pops off the first element from the queue every set amount of time assigned by the methods outlined.

2.1 Results

Figure 3 compares the number of unresolved SYN packets in the queues from the SYN flooding attack using three different ways to set the server time-out. The figure shows us that when using no server time-out to dequeue some of the unresolved

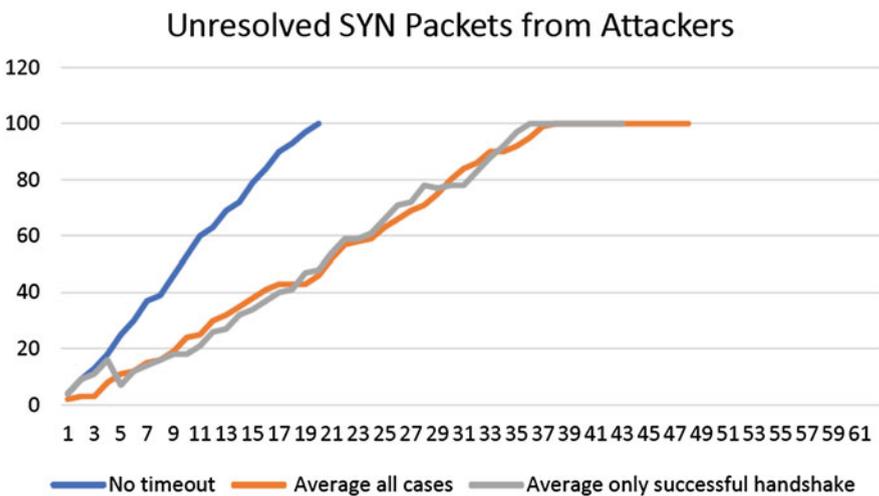


Fig. 3 TCP SYN Flood Attack

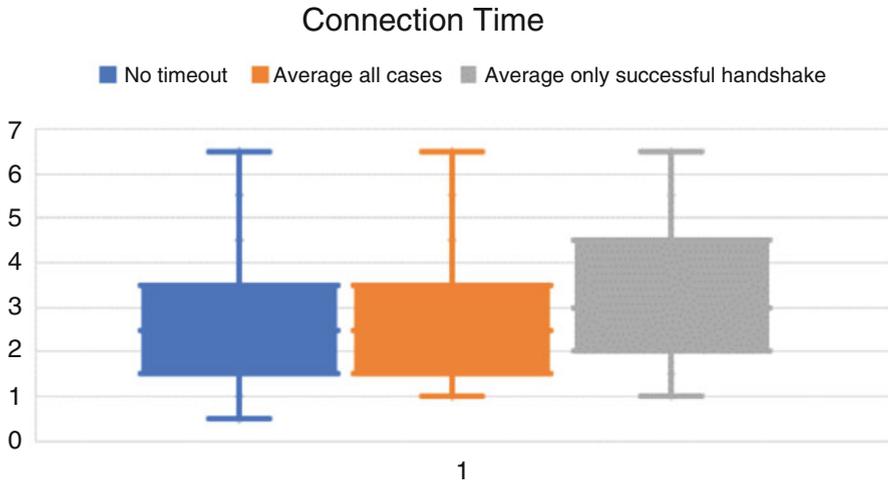


Fig. 4 Connections Times for all IPs

request, it can lead to a steady increase in the numbers of unresolved requests, leading to faster DDoS success for the attacker. It actually reached the peak so fast that the server crashed before half of the other methods had even finished. The interesting comparison is between the other two methods. When the server time-out is applied as all connections average, we see lots of sharp decrease and parts where it is stabilized, and it actually reached the maximum a little bit slower than other methods. On the other hand, the method to apply only successful handshakes does not differ that much from the average all cases method, except for the fact that it reached maximum quicker.

Figure 4 compares the connection times in the server using three different ways to set the server time-out. The figure shows us that when using no server time-out to dequeue some of the unresolved request, the connection time throughout is almost the same as when the server time-out is applied as all connections average. On the other hand, the method to apply only successful handshakes yield a pretty significantly longer connection time, which is due to the longer time it takes to dequeue the connections, but it's also due to the fact that we included the actual transmission of the data, which will take longer compared to SYN flood attacks only under our stimulation environment.

3 Conclusion and Future Work

Through the comparison between the three different methods, we can say that under these circumstances of the stimulation environment, the method to dequeue every average time of all the successful handshakes and transmission would be the most

ideal. It shows that we can transmit more files than not using a time-out method and steadier than using an average of all cases including the attack times. For future work, the environment would be more beneficial if each of the IP is dynamically timed, so that we can include a more accurate connection times of the cases that were attacking our server.

References

1. E. Beccaria, V. Bogado, J.A. Palombarini, A devs-based simulation model for biogas generation for electrical energy production, in *2018 IEEE Biennial Congress of Argentina (ARGENCON)* (IEEE, 2018), pp. 1–8
2. M. Bellaïche, J.-C. Grégoire, Syn flooding attack detection by tcp handshake anomalies. *Secur. Commun. Netw* **5**(7), 709–724 (2012)
3. Cloudflare, Wikipedia, <https://en.wikipedia.org/wiki/Cloudflare>. Accessed 15 June 2021
4. DDoS, Ddos attacks target multiple games including final fantasy XIV, Assassin's Creed. October 9, 2018. <https://www.scmagazine.com/news/cybercrime/ddos-attacks-target-multiple-games-including-final-fantasy-xiv>. Accessed 15 June 2021
5. W. Ding, S. Zhang, Z. Zhao, A collaborative calculation on real-time stream in smart cities. *Simul. Model. Pract. Theory* **73**, 72–82 (2017)
6. M. Etemad, M. Aazam, M. St-Hilaire, Using devs for modeling and simulating a fog computing environment, in *2017 International Conference on Computing, Networking and Communications (ICNC)* (IEEE, 2017), pp. 849–854
7. B. Hang, R. Hu, A novel syn cookie method for tcp layer ddos attack, in *2009 International Conference on Future Biomedical Information Engineering (FBIE)* (IEEE, 2009), pp. 445–448
8. O. Kupreev, E. Badovskaya, A. Gutnikov, Ddos attacks in q1 2019 (2019)
9. J. Lemon, et al. Resisting syn flood dos attacks with a syn cache. In *BSDCon*, vol. 2002, pp. 89–97 (2002)
10. S. Mirzaie, A.K. Elyato, M.A. Sarram, Preventing of syn flood attack with iptables firewall, in *2010 Second International Conference on Communication Software and Networks* (IEEE, 2010), pp. 532–535
11. SYN flood, Wikipedia. https://en.wikipedia.org/wiki/SYN_flood. Accessed 15 June 2021
12. Y. Van Tendeloo, H. Vangheluwe, An evaluation of devs simulation tools. *Simulation* **93**(2), 103–121 (2017)
13. Differences between TCP and UDP. July 13, 2021. <https://www.geeksforgeeks.org/differences-between-tcp-and-udp/>. Accessed 1 August 2021
14. B.P. Zeigler, T.G. Kim, H. Praehofer, *Theory of Modeling and Simulation* (Academic press, 2000)

Part II
Computational Intelligence, Data Science,
HPC, Optimization and Applications

Dielectric Polymer Genome: Integrating Valence-Aware Polarizable Reactive Force Fields and Machine Learning



Kuang Liu, Antonina L. Nazarova, Ankit Mishra, Yingwu Chen, Haichuan Lyu, Longyao Xu, Yue Yin, Qinai Zhao, Rajiv K. Kalia, Aiichiro Nakano, Ken-ichi Nomura, Priya Vashishta, and Pankaj Rajak

1 Introduction

Development of high-performance dielectric polymers is urgently needed for the advancement of a wide range of energy technologies, including energy-storage and pulsed-power technologies [1–3]. Recent advent of materials genome (i.e., applying informatics to design new materials significantly faster than the conventional trial-and-error approach) [4] has enabled the rational design of dielectric polymers [3], thereby heralding an exciting era of polymer genome [5]. A major performance indicator of dielectric polymer is dielectric constants. While the dielectric genome has previously been proposed for inorganic crystals [6], high computational costs

K. Liu · A. Mishra · R. K. Kalia · A. Nakano · K.-i. Nomura (✉) · P. Vashishta
Collaboratory of Advanced Computing and Simulations, Department of Computer Science,
Department of Physics & Astronomy, Department of Chemical Engineering & Materials Science,
Department of Biological Sciences, University of Southern California, Los Angeles, CA, USA
e-mail: liukuang@usc.edu; ankitmis@usc.edu; rkalia@usc.edu; anakano@usc.edu;
knomura@usc.edu; priyav@usc.edu

A. L. Nazarova
Loker Hydrocarbon Research Institute and The Bridge@USC, Department of Chemistry,
University of Southern California, Los Angeles, CA, USA
e-mail: nazarova@usc.edu

Y. Chen · H. Lyu · L. Xu · Y. Yin · Q. Zhao
Mork Family Department of Chemical Engineering & Materials Science, University of Southern
California, Los Angeles, CA, USA
e-mail: yingwuch@usc.edu; haichuan@usc.edu; longyaox@usc.edu; yueyin@usc.edu;
qinaizha@usc.edu

P. Rajak
Argonne Leadership Computing Facility, Argonne National Laboratory, Lemont, IL, USA
e-mail: prajak@anl.gov

of quantum-mechanically evaluating dielectric constants [7–9] make it still a challenge. The situation is even worse for polymers, where computing is far more expensive since complex chemical and morphological features essentially dictate their dielectric constants. As a result, the highly anticipated “dielectric polymer genome” still remains elusive.

This computational challenge may be addressed by recent developments in first principles-informed reactive molecular dynamics (RMD) simulations based on polarizable reactive force fields (ReaxPQ) [10–12]. The ReaxPQ model, based on a polarizable charge equilibration (PQEq) scheme, has significantly improved the accuracy for predicting dielectric constants in orders-of-magnitude shorter computational times compared with quantum-mechanical (QM) calculations. In this paper, we further improve the predictive power of ReaxPQ by introducing a new valance (or charge state)-aware ReaxPQ (ReaxPQ-v) model, thereby achieving near-quantum accuracy for predicting dielectric constants of polymers. We incorporate ReaxPQ-v into our high-throughput computational synthesis framework of polymers [13]. We thereby construct a large dataset of structurally diverse 1276 dielectric polymers with quantum accuracy, which represents a frontier of dielectric polymer genome research.

In order to explore the large combinatorial design space of dielectric polymers, the above approach alone is not sufficient. It is essential to further exploit recent advancements in machine-learning (ML) approaches to molecular and materials sciences [14], in particular deep learning (DL) [15]. For example, various artificial neural networks have been used successfully to identify structure–property relationships in materials, including graph neural networks (GNNs) [16, 17] and recurrent neural networks (RNNs) [18]. Here, it is notable that a wide availability of public ML tools has now democratized ML tasks that could utilize the above-mentioned state-of-the-art dataset. As a proof of concept of the use of our new dielectric polymer dataset by a broad scientific community, we have used the dataset in two graduate courses, MASC 599 (Basics of Machine Learning for Materials) and PHYS 516 (Methods of Computational Physics), where ML models of progressive complexity—multilayer perceptron (MLP), random forest, and RNN—were applied. In combination with the ReaxPQ-v based computational framework, MLP and random forest models have achieved decent accuracy for predicting the dielectric constants of polymers, while the RNN model is under tuning.

2 Method

This section describes the ReaxPQ-v model, computational framework for dataset generation, and ML models.

2.1 *Valance-Aware ReaxPQ (ReaxPQ-v) Model*

Reactive molecular dynamics (RMD) simulation based on first principles-informed reactive force fields (ReaxFF) [19–21] significantly reduces the computational cost of quantum-mechanically simulating chemical reactions in materials. A recently proposed polarizable reactive force field (ReaxPQ) model extends ReaxFF by accurately describing dielectric properties based on a polarizable charge model, in which each atom consists of core and shell charges that are connected by a spring [10–12]. Unfortunately, advanced dielectric polymers contain atomic species at various charge states (or valences), making accurate prediction of dielectric constants very hard. Here, we introduce a valence-aware ReaxPQ (ReaxPQ-v) model to accurately describe the dielectric response of amorphous polymers based on the quantum-mechanically informed atomic polarizability. We first determine the polarizability of constituent atomic species quantum-mechanically, using the Hartree–Fock method with split valence polarization with diffuse function (def2-SVPD) basis sets [22]. These polarizability values for different valences are then used to estimate the spring constants between core and shell charges. The new ReaxPQ-v model has been implemented in our scalable parallel RMD simulation engine called RXMD [23].

2.2 *Computational Framework and Dataset Generation*

We generate amorphous polymer structures using our computational synthesis framework [13]. The framework first generates a single polymer chain using a Simplified Molecular Input Line Entry System (SMILES) [24] string and Open Babel software [25]. Multiple polymer chains are then placed at sufficiently large distances from each other in a simulation box. The RMD simulation system is subjected to a number of consolidation and relaxation steps, until the system reaches a desired density. Following this procedure, we constructed a synthetic dataset of structurally diverse 1276 polymers. Dielectric constants of these polymers are then computed with the ReaxPQ-v model, using a high-throughput, parallel RMD simulation workflow. The new dataset composed of a large number of amorphous polymer structures, augmented with quantum-accuracy dielectric constants, provides an indispensable test ground for developing predictive ML models of polymer dielectric constants.

2.3 Machine-Learning Models

To predict the dielectric-constant values from SMILES inputs, we have thus far tested three ML models: multilayer perceptron (MLP), random forest, and recurrent neural network (RNN).

The MLP algorithm has played a fundamental role in the evolution of artificial neural networks and ML at large [26]. Pioneered by Rosenblatt [27], MLP has been transformed from a binary classifier to an affordable method of solving supervised estimation problems [28]. Here, we use a two hidden-layer perceptron estimator, where each layer is comprised of 1136 neurons. Backpropagation is employed for weight adjustment, whereas the activation function of modified rectified linear unit (RELU) demonstrated the best fit.

To develop a random forest model, we use the Citrination online platform (<http://citrination.com>) for feature generation and model construction. Citrination provides autogenerated features that can be used to develop a model or argument on the customized features provided by users. This is particularly useful for developing an ML model for materials, in which each entry is commonly represented by the chemical formula or SMILES string. Given the training data explained above, we first generate over 100 features (63 standard and 71 extended features) based on the given chemical formula and/or SMILES strings. The standard set consists of elemental properties, molecule features, and analytic features. The extended set contains elemental properties and molecule features. These features include electronic configurations of constituent atoms (e.g., the number of valence electrons), which are readily available from the periodic table. In addition, quantum-mechanical calculations in the framework of density functional theory (DFT) [29–32] are used to produce additional data such as the electronic energy. We also generate 26 additional features based on functional groups in the polymer dataset and compare the model performance.

Furthermore, Citrination can identify a small subset of features that contribute mostly to model prediction, thereby achieving significant dimensionality reduction. Table 1 shows four most important features sorted by their contribution percentage to model development, in the case of random forest (see the next section).

Given the training data and features as described above, we next select an ML model. Our choice in Citrination is random forest, which corrects the overfitting tendency of decision trees [33]. Random forest constructs an ensemble of tree predictors at training time. Individual trees sample random vector values independently of each other, but with the same distribution for all trees in the forest.

Table 1 Most important features

Feature	Contribution (%)
Mean of DFT energy density	42
Mean of total number of valence electrons	30
Mean of row in the periodic table	12
Mean of number of p valence electrons	9

In addition to MLP and random forest, we also consider RNN, in which the next run input is dependent on the previous one, while communicating through hidden and context layers [34]. RNN is commonly used for problems involving data sequences such as text or symbol arrays. The basic organization of RNN involves three interconnected layers: input, output, and hidden nodes. The context units are propagated from the output of the hidden layers and recurrently connected. Among themselves, nodes communicate via direct synapses represented with the weighing parameters. Due to dynamic behavior, RNN modules employ commonly used activation functions such as sigmoid and RELU. During the training step, the value of the hidden layer $h_v^{(t)} \in \mathbf{R}^m$ can be derived as

$$h_v^{(t)} = f \left(\mathbf{W}x^{(t)} + \mathbf{U}h_v^{(t-1)} + b^h \right), \quad (1)$$

where $x^{(t)}$ is an input vector at step t , \mathbf{W} are the weighing parameters for a hidden layer, \mathbf{U} are the weighing parameters for a context layer, $b^h \in \mathbf{R}^m$ are the bias parameters of RNN model, and f is an activation function. In the traditional RNN, the message function $y_v^{(t)} \in \mathbf{R}^l$ is

$$y_v^{(t)} = \mathbf{V}h_v^{(t)} + b^c, \quad (2)$$

where \mathbf{V} are the weighing parameters for output and $b^c \in \mathbf{R}^l$ denotes bias.

In 1977, Hochreiter and Schmidhuber developed a long short-term memory (LSTM) unit to avoid the vanishing and exploding gradient issues that accompany traditional RNNs [35]. Subsequently, Cho et al. developed a new RNN model called the gated recurrent unit (GRU), which employs the same basic principles as LSTM [36]. Both architectures use gating units denoting the further outcome of the next hidden layer $h_v^{(t+1)}$ and computation of $y_v^{(t)}$. However, GRU architecture implements lower number of filters and operations for inner cycle $h_v^{(t)}$ calculations. For GRU implementation, updated gate $z_v^{(t)}$ and reset gate $r_v^{(t)}$ can be denoted as

$$z_v^{(t)} = \sigma \left(\mathbf{W}^z (n) x^{(t)} + \mathbf{U}^z (m) h^{(t-1)} \right), \quad (3)$$

$$r_v^{(t)} = \sigma \left(\mathbf{W}^r (n) x^{(t)} + \mathbf{U}^r (m) h^{(t-1)} \right), \quad (4)$$

where n and m are tensor dimensions.

Output value $h_v^{(t)}$ is computed on the basis of the interim $\tilde{h}_v^{(t)}$. These gate outputs are accountable for the information to be left out or excluded from the training:

$$\tilde{h}^{(t)} = \text{tanh} \left(\mathbf{U}_h (m) \left(r_v^{(t)} \odot h^{(t-1)} \right) + \mathbf{W}_h (n) x^{(t)} \right), \quad (5)$$

representing both update gate and interim values,

$$h^{(t)} = z^{(t)} \odot h^{(t)} + (1 - z^{(t)}) \odot \tilde{h}^{(t-1)}. \quad (6)$$

For the present polymer problem, the initial states h_v^o are set to the input features of oligomers, such as structure and dielectric-constant (ϵ) values. Due to their efficiency and robustness, both LSTM and GRU network architectures have become essential computational tools in various areas, including the study of structure–activity relationships in life and materials sciences [37].

3 Results and Discussion

This section describes the training dataset and ML model prediction results.

3.1 Training Dataset

Continuous enlargement of the training set is essential for improving the robustness, accuracy, and efficiency of ML models. For the current study, a data-driven framework is implemented for a synthetic polymer dataset incorporating various functional groups [13]. In total, a diverse library of 1276 oligomeric structures with unknown chirality is computationally generated and structurally characterized (Fig. 1). The training data consist of the unique SMILES representation (with no chiral specificity) and the dielectric constant, ϵ_∞ , calculated by ReaxPQ-v for computationally synthesized amorphous polymer structures. This is an important property as the relative polarity of the polymers determines its permittivity. The pool of SMILES fingerprints consists mainly of separate or fused heterocycles as well as of functionalized parent moieties. Electronically and sterically diverse substituents are incorporated to capture all potential features controlling ϵ_∞ .

3.2 Predictive Accuracy of Machine Learning

We first use the MLP model and run prediction for the entire training set of polymer samples. For efficient training, the SMILES input is transformed to binary representation together with zero padding. Polymers, whose dielectric constants are less than 1.5, are removed from the training set (Fig. 2a), as these statistical outliers do not follow the general trend. Additionally, the input representations are examined for the presence of duplicated SMILES entries, leading to the total number of the training sets to be 828 (Fig. 2b).

Figure 3 compares the parity plot (i.e., model prediction vs. input dataset) of three developed MLP models for an MLP model, using different activation functions:

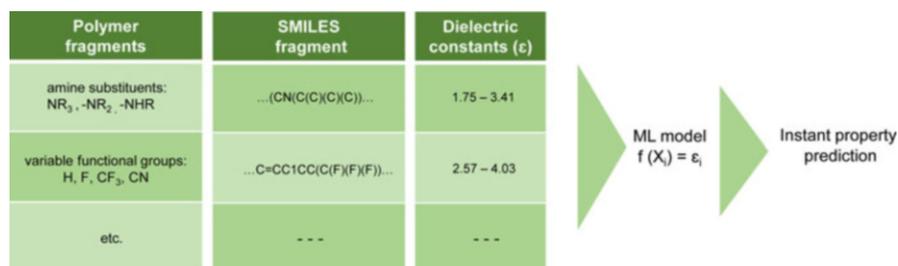


Fig. 1 Schematic representation of the key elements of the ML model, which include fingerprinting, learning, and prediction

Fig. 2 Dielectric constants (a) and SMILE dimensions (b) of the polymer dataset

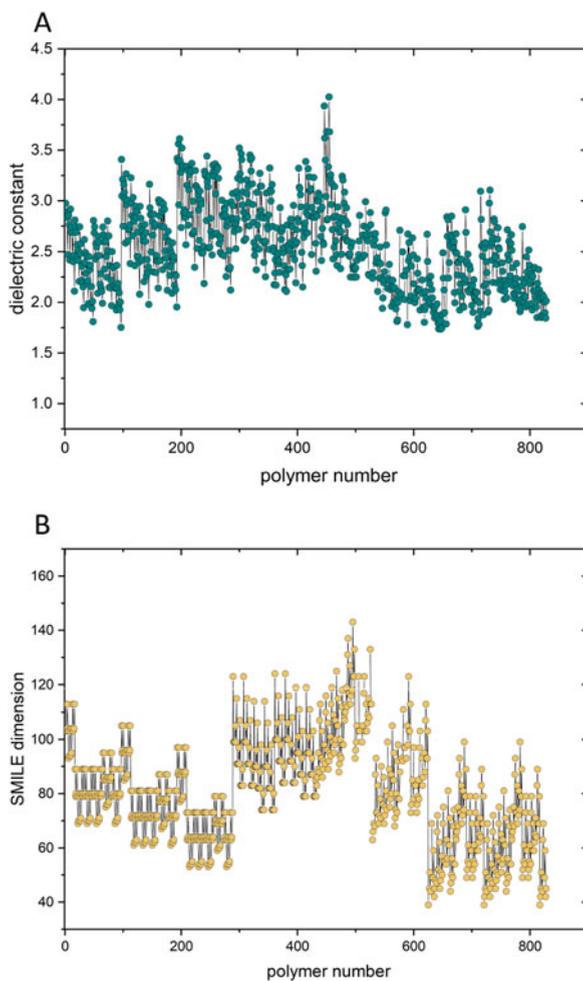
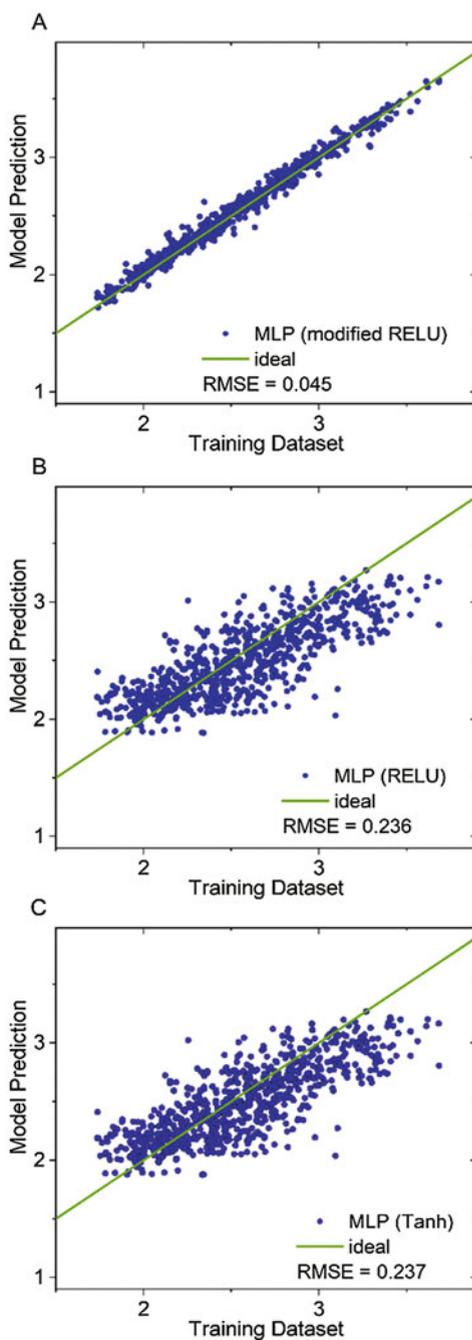
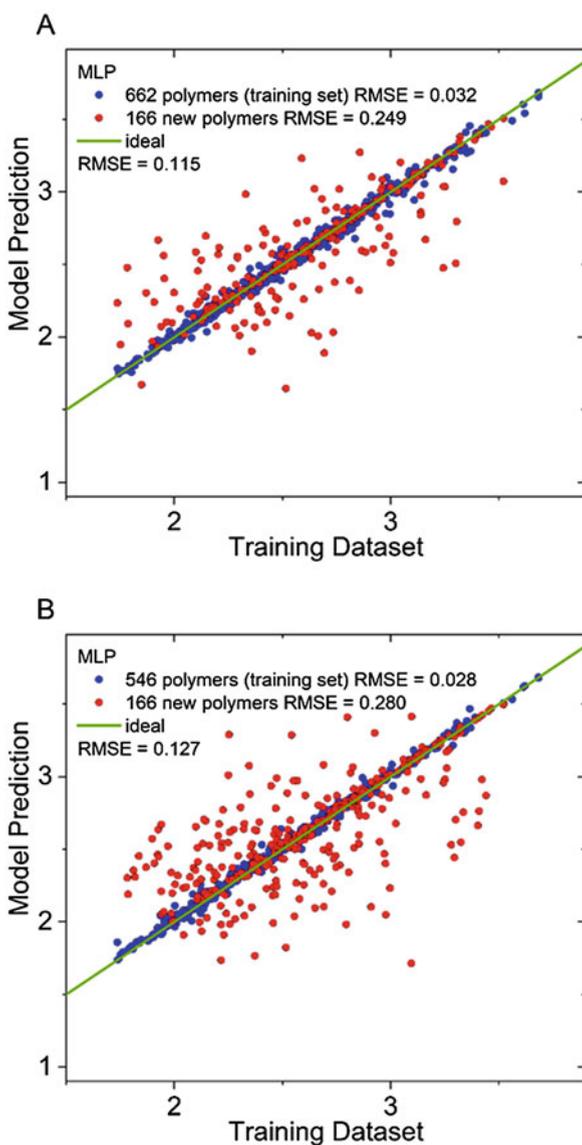


Fig. 3 Model prediction vs. training data of polymer dielectric constants using the MLP model with variable activation functions: (a) modified RELU (RMSE = 0.045); (b) standard RELU (RMSE = 0.236); and (c) Tanh (RMSE = 0.237)



(A) modified RELU activation function, in which the standard RELU is used for gradient calculation within the backpropagation learning rule, whereas binary step is used for activation itself; (B) standard RELU activation function; and (C) hyperbolic tangent (Tanh) function. Out of the three, the one obtained with modified RELU activation function demonstrates an excellent agreement between the first principles-derived dielectric-constant values with the model-predicted ones. The root mean squared error (RMSE) is 0.045.

Fig. 4 Performance of an MLP prediction model (Elman type). Comparison of prediction on (a) 662 polymer and (b) 546 polymer training sets



The experiment with the variable training sets demonstrates how the enlargement of the sample pool affects the overall MLP performance. Figure 4 shows a parity plot of the MLP model for 662 and 546 training polymer datasets. In the RMSE metrics, the training set of 667 samples demonstrates higher performance than the set of 546 samples. However, the individual RMSE values indicate the predictive performance within the training pool to be expectedly better for the low volume set (0.028 for 546 versus 0.032 for 662 polymers).

As explained in the previous section, we have built a random forest model using the Citrination online platform. Prior to model development, we cleaned the polymer dielectric dataset by removing unphysical records. We obtained 846 data points in total. The random forest model [33] is implemented using the Lolo library, where a variety of data types and ML functions are supported, including continuous and categorical features, regression and classification trees, jackknife variance estimates, hyperparameter optimizations, validation metrics for accuracy, and uncertainty quantification. Jackknife is an uncertainty estimate method by resampling data, similar to the leave-one-out approach for cross-validation. Suppose $X = (X_1, X_2, X_3, \dots, X_n)$ is a set of observed samples. Jackknife i th set of samples is defined as the original set without the i th sample,

$$X_{[i]} = \{X_1, X_2, \dots, X_{i-1}, X_{i+1}, \dots, X_n\} \quad (7)$$

The estimated value $\theta_{(i)}$ by an estimator S is obtained as

$$\theta_{(i)} = S(X_{[i]}) \quad (8)$$

The performance of a model is evaluated by the bias and standard error of the estimator. Table 2 shows hyperparameters that are used to build our model.

Figure 5 shows the resulting parity plot with uncertainty in the predicted value. Data points that are within the input feature space are labeled as “not extrapolating,” otherwise labeled as “extrapolating.”

Table 3 summarizes various model-performance metrics, such as RMSE, nondimensional model error (NDME), standard error in the estimate of RMSE (SE-RMSE), and standard error in the estimate of NDME (SE-NDME). Here, NDME is the ratio between the RMSE and standard deviation. The small RMSE and NDME values indicate that the developed model possesses a good prediction capability. We examine three sets of features: (1) SMILES string; (2) 26 additional features that consist of polymer functional groups; and (3) SMILES string and the functional

Table 2 Hyperparameters used in our random forest model

Number of estimators	155
Minimum samples per leaf	1
Maximum tree depth	30
Leaf model	Mean
Number of cross-validation folds	3

Fig. 5 Model prediction vs. training data of polymer dielectric constants using the random forest model

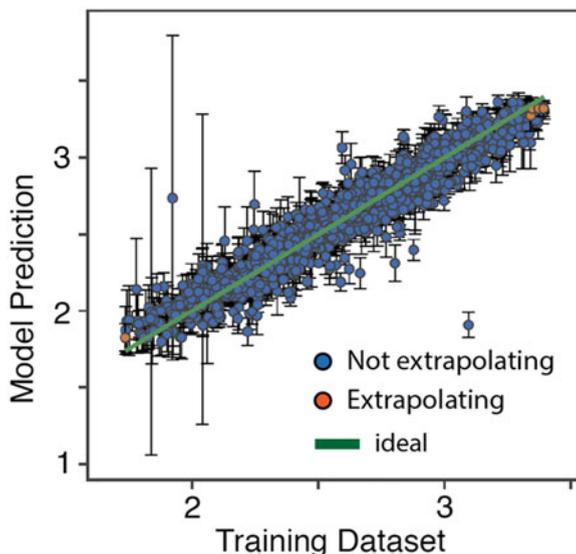


Table 3 Performance metrics of the random forest model

Features	RMSE	SE-RMSE	NDME	SE-NDME
SMILES	0.129	0.00713	0.327	0.0180
Functional groups	0.174	0.00281	0.439	0.00709
SMILES & functional groups	0.130	0.00330	0.329	0.00833

group together. Use of either SMILES string or the functional group feature alone results in a similar RMSE around 0.13. However, we found that SE-RMSE substantially improves by 53% when combining the SMILES and the functional group features together.

While the optimization of RNN model is still in progress, Fig. 6 shows that the loss function (which represents the deviation of model prediction from ground truth) is reduced monotonically as a function of model training epochs.

4 Conclusion

We have developed a high-throughput simulation-ML framework for computationally synthesizing a large dataset of polymers, evaluating their dielectric constants, and learning structure–property relationships using a new valence-aware polarizable reactive force-field model and ML models including multilayer perceptron, random forest, and recurrent neural network. A high level of prediction accuracy was achieved by a large size of the simulated training dataset. Employing the full dataset

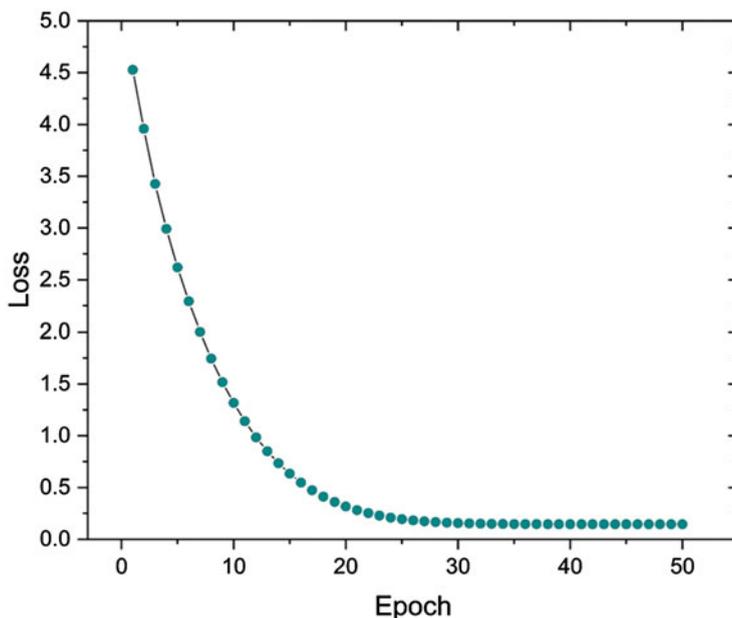


Fig. 6 Convergence of the RNN model performance as a function of training epochs

of structurally diverse 1276 polymers, the ML models have predicted the dielectric constant of polymers with decent accuracy.

As an extension of this work, we are currently developing a graph neural network model [16, 17]. Furthermore, these regression models are being combined with our active learning framework [38], so as to predict the optimal polymer structure with minimal number of first principles-based calculations. This is achieved by incrementally expanding the training set, while balancing exploration and exploitation based on Bayesian optimization. Finally, we are implementing all simulation and ML models on scalable parallel computing platforms [11, 39–41], including the United States’ first exaflop/s computer, Aurora A21, to be installed in 2021, under our Aurora Early Science Program award [42]. Such scalable ML prediction models, once optimized and validated, will be indispensable for further enlarging the search space for superior dielectric polymers by orders-of-magnitude.

Acknowledgments This work was supported by the Office of Naval Research through a Multidisciplinary University Research Initiative (MURI) Grant N00014-17-1-2656. The simulations were performed at the Argonne Leadership Computing Facility under the DOE INCITE and Aurora Early Science programs and at the Center for High Performance Computing of the University of Southern California.

References

1. Q.M. Zhang, V. Bharti, X. Zhao, Giant electrostriction and relaxor ferroelectric behavior in electron-irradiated poly(vinylidene fluoride-trifluoroethylene) copolymer. *Science* **280**(5372), 2101–2104 (1998). <https://doi.org/10.1126/science.280.5372.2101>
2. B.J. Chu et al., A dielectric polymer with high electric energy density and fast discharge speed. *Science* **313**(5785), 334–336 (2006). <https://doi.org/10.1126/science.1127798>
3. V. Sharma et al., Rational design of all organic polymer dielectrics. *Nat. Commun.* **5**, 4845 (2014). <https://doi.org/10.1038/ncomms5845>
4. A. Jain, K.A. Persson, G. Ceder, Research update: the materials genome initiative: Data sharing and the impact of collaborative ab initio databases. *APL Mater* **4**(5) (2016). <https://doi.org/10.1063/1.4944683>
5. C. Kim, A. Chandrasekaran, T.D. Huan, D. Das, R. Ramprasad, Polymer genome: a data-powered polymer informatics platform for property predictions. *J. Phys. Chem. C* **122**(31), 17575–17585 (2018). <https://doi.org/10.1021/acs.jpcc.8b02913>
6. K. Andersen, S. Latini, K.S. Thygesen, Dielectric genome of van der Waals heterostructures. *Nano Lett.* **15**(7), 4616–4621 (2015). <https://doi.org/10.1021/acs.nanolett.5b01251>
7. P. Umari, A. Pasquarello, Ab initio molecular dynamics in a finite homogeneous electric field. *Phys. Rev. Lett.* **89**(15), 157602 (2002). <https://doi.org/10.1103/PhysRevLett.89.157602>
8. I. Souza, J. Iniguez, D. Vanderbilt, First-principles approach to insulators in finite electric fields. *Phys. Rev. Lett.* **89**(11), 117602 (2002). <https://doi.org/10.1103/PhysRevLett.89.117602>
9. S. Fukushima et al., Effects of chemical defects on anisotropic dielectric response of polyethylene. *AIP Adv.* **9**(4), 045022 (2019). <https://doi.org/10.1063/1.5093566>
10. S. Naserifar, D.J. Brooks, W.A. Goddard, V. Cvacek, Polarizable charge equilibration model for predicting accurate electrostatic interactions in molecules and solids. *J. Chem. Phys.* **146**(12), 124117 (2017). <https://doi.org/10.1063/1.4978891>
11. K. Liu et al., Shift-collapse acceleration of generalized polarizable reactive molecular dynamics for machine learning-assisted computational synthesis of layered materials. *Proc ScalA* **18**, 41–48., IEEE (2018). <https://doi.org/10.1109/ScalA.2018.00009>
12. Y. Li et al., Scalable reactive molecular dynamics simulations for computational synthesis. *Comput. Sci. Eng.* **21**(5), 64–75 (2019). <https://doi.org/10.1109/MCSE.2018.110150043>
13. A. Mishra et al., Computational framework for polymer synthesis to study dielectric properties using polarizable reactive molecular dynamics. *ACS Central Sci.*, submitted (2020)
14. K.T. Butler, D.W. Davies, H. Cartwright, O. Isayev, A. Walsh, Machine learning for molecular and materials science. *Nature* **559**(7715), 547–555 (2018). <https://doi.org/10.1038/s41586-018-0337-2>
15. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. *Nature* **521**(7553), 436–444 (2015). <https://doi.org/10.1038/nature14539>
16. D. Duvenaud et al., Convolutional networks on graphs for learning molecular fingerprints. *Proc. NeurIPS 2015* **28** (2015)
17. K. Liu, K. Nomura, P. Rajak, R.K. Kalia, A. Nakano, P. Vashishta, Graph neural network analysis of layered material phases. *Proc. SpringSim-HPC 2019, SCS* (2019)
18. M.H.S. Segler, T. Kogej, C. Tyrchan, M.P. Waller, Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Sci.* **4**(1), 120–131, 2018. <https://doi.org/10.1021/acscentsci.7b00512>
19. A.C.T. van Duin, S. Dasgupta, F. Lorant, W.A. Goddard, ReaxFF: A reactive force field for hydrocarbons. *J. Phys. Chem. A* **105**(41), 9396–9409, 2001. <https://doi.org/10.1021/jp004368u>
20. A. Nakano et al., De novo ultrascale atomistic simulations on high-end parallel supercomputers. *Int. J. High. Perform. Comput. Appl.* **22**(1), 113–128 (Feb 2008). <https://doi.org/10.1177/1094342007085015>

21. T.P. Senftle et al., The ReaxFF reactive force-field: Development, applications and future directions. *npj Comput. Mater.* **2**, 15011 (2016). <https://doi.org/10.1038/npjcompumats.2015.11>
22. A. Hellweg, D. Rappoport, Development of new auxiliary basis functions of the Karlsruhe segmented contracted basis sets including diffuse basis functions (def2-SVPD, def2-TZVPPD, and def2-QVPPD) for RI-MP2 and RI-CC calculations. *Phys. Chem. Chem. Phys.* **17**(2), 1010–1017 (2015). <https://doi.org/10.1039/C4CP04286G>
23. K. Nomura, R.K. Kalia, A. Nakano, P. Rajak, P. Vashishta, RXMD: A scalable reactive molecular dynamics simulator for optimized time-to-solution. *SoftwareX* **11**, 100389 (2020). <https://doi.org/10.1016/j.softx.2019.100389>
24. D. Weininger, SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Info. Comp. Sci.* **28**, 31–36 (1988). <https://doi.org/10.1021/ci00057a005>
25. N.M. O’Boyle, M. Banck, C.A. James, C. Morley, T. Vandermeersch, G.R. Hutchison, Open Babel: An open chemical toolbox. *Aust. J. Chem.* **3**(1), 33 (2011). <https://doi.org/10.1186/1758-2946-3-33>
26. H. Ramchoun, M.A.J. Idrissi, Y. Ghanou, M. Ettaouil, Multilayer perceptron: Architecture optimization and training. *IJIMAI* **4**(1), 26–30 (2016). <https://doi.org/10.9781/ijimai.2016.415>
27. F. Rosenblatt, *Principles of Neurodynamics: Perceptions and the Theory of Brain Mechanisms* (Spartan, Washington, DC, 1962)
28. N. Talebi, A.M. Nasrabadi, I. Mohammad-Rezazadeh, Estimation of effective connectivity using multi-layer perceptron artificial neural network. *Cogn. Neurodyn.* **12**(1), 21–42 (2018). <https://doi.org/10.1007/s11571-017-9453-1>
29. P. Hohenberg, W. Kohn, Inhomogeneous electron gas. *Phys. Rev.* **136**(3B), B864–B871 (1964). <https://doi.org/10.1103/PhysRev.136.B864>
30. R.M. Martin, *Electronic Structure: Basic Theory and Practical Methods* (Cambridge University Press, Cambridge, UK, 2008)
31. F. Shimojo et al., A divide-conquer-recombine algorithmic paradigm for multiscale materials modeling. *J. Chem. Phys.* **140**(18), 18A529 (2014). <https://doi.org/10.1063/1.4869342>
32. F. Shimojo et al., QXMD: An open-source program for nonadiabatic quantum molecular dynamics. *SoftwareX* **10**, 100307 (2019). <https://doi.org/10.1016/j.softx.2019.100307>
33. L. Breiman, Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
34. Z.C. Lipton, J. Berkowitz, C. Elkan, A critical review of recurrent neural networks for sequence learning. *arXiv*, 1506.00019v4 (2015)
35. S. Hochreiter, J. Schmidhuber, Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997). <https://doi.org/10.1162/neco.1997.9.8.1735>
36. K. Cho et al., Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Proc. EMNLP* **2014** (2014)
37. A. Cherkasov et al., QSAR modeling: Where have you been? Where are you going to? *J. Med. Chem.* **57**(12), 4977–5010 (2014). <https://doi.org/10.1021/jm4004285>
38. L. Bassman et al., Active learning for accelerated design of layered materials. *npj Comput. Mater.* **4**, 74 (2018). <https://doi.org/10.1038/s41524-018-0129-0>
39. P. Rajak et al., Neural network molecular dynamics at scale. *Proc. ScaDL*, accepted (2020), IEEE, 2020
40. N.A. Romero et al., Quantum molecular dynamics in the post-petaflops era. *IEEE Computer* **48**(11), 33–41 (2015)
41. S. C. Tiwari et al., Quantum Dynamics at Scale: Ultrafast Control of Emergent Functional Materials, *Proc HPCAsia2020*, Best Paper Award, Jan 15 ACM, 2020. <https://doi.org/10.1145/3368474.3368489>
42. R. F. Service, Design for US exascale computer takes shape. *Science* **359**(6376), 617–618 (2018). <https://doi.org/10.1126/science.359.6376.617>

A Methodology to Boost Data Science in the Context of COVID-19



Carlos J. Costa and Joao Tiago Aparicio

1 Introduction

We may define data science as the interception of computer science (CS), domain knowledge (DK), and mathematics (MT) [1]. It includes techniques developed in some traditional fields like artificial intelligence, statistics, or machine learning. But it has an application in a specific context. This context is a specific domain of knowledge. And so, it is essential to employ a methodology that may contribute to the improvement of the knowledge creation outputs. It is in this context crucial to identify possible approaches. By analyzing the existing methodologies, it is essential to enlarge the scope of knowledge discovery or data mining. Almost all the approaches emphasize the process. The following sections describe KDD (knowledge discovery in databases), CRISP-DM (cross-industry standard process for data mining), SEMMA (sampling, exploring, modifying, modeling, and assessing), ASUM (Analytics Solutions Unified Method), and TDSP (Team Data Science Process). Then a new approach POST-DS (Process Organization and Scheduling electing Tools for Data Science) is presented. The next section illustrates this methodology usage.

C. J. Costa (✉)

ISEG, Universidade de Lisboa, Lisboa, Portugal

e-mail: carlos.costa@acm.org

J. T. Aparicio

Instituto Superior Tecnico, Universidade de Lisboa, Lisboa, Portugal

e-mail: joao.aparicio@tecnico.ulisboa.pt

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_7

2 Literature Review

KDD (knowledge discovery in databases) describes the overall process of discovering useful knowledge from data. Data mining is the usage of specific algorithms for extracting patterns from data. So, data mining refers to an appropriate step in this process. KDD is the nontrivial process of finding valid, new, possibly useful, and ultimately understandable patterns in data. The KDD process is iterative and interactive, involving numerous steps with many decisions made by the analyst. It is essential in developing an understanding of the data, creating a target dataset, cleaning, and preprocessing. Then, several tasks must be performed, like data reduction and projection. The analyst also has to match the goals of the KDD process to a particular data mining method, exploratory analysis, and model and hypothesis selection. An essential task is interpreting mined patterns and using the knowledge directly [21].

The first step is developing an initial interpretation in the context of the domain. It consists of the relevant prior knowledge and the goal identification of the KDD process from the customers' perspective.

The second step consists of creating a target data set by selecting a dataset or focusing on a subset of variables.

The third step consists of data cleaning and preprocessing. Basic operations may include removing noise, collecting the necessary information to model or account for noise, deciding on strategies for handling missing values, and accounting for time sequence information and known changes.

The fourth step involves data reduction and projection. It consists of finding useful features to represent the data. The adequate number of variables can be obtained, with dimensional reduction or transformation methods. Invariant representations for the data may also be identified.

The fifth step consists of matching the goals of the KDD process to a particular data mining method. Those methods may be summarization, classification, regression, or clustering.

The sixth step consists of exploratory analysis and model and hypothesis selection. It involves choosing the data mining algorithm(s) and selecting method(s) to be used for searching for data patterns. This process comprises deciding which models and parameters might be appropriate (Fig. 1).

The seventh step consists of data mining. It includes searching for patterns of interest in a particular representational form or a set of such representations. Examples are classification rules or trees, regression, and clustering. The user can significantly aid the data mining method by correctly performing the preceding steps.

The eighth step consists of interpreting mined patterns. In this step, it is possible to return to any of the previous steps for additional iteration. This step can also include visualization of the extracted patterns and models or display of the data given the obtained models.

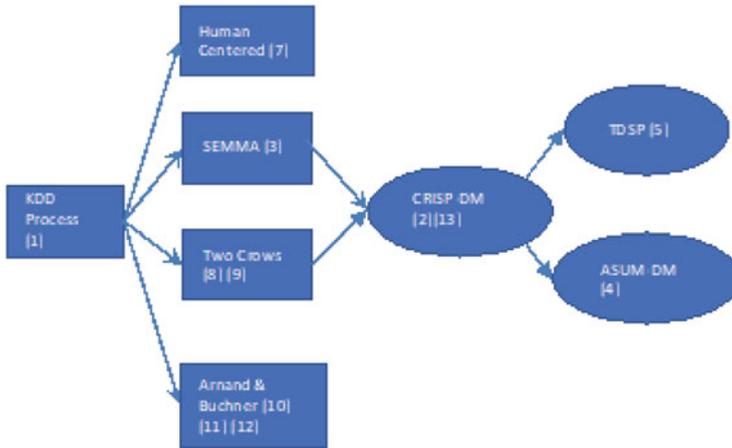


Fig. 1 Evolution of data mining process models and methodologies [7–13]

The ninth step consists of acting on the discovered knowledge. This step consists of using the knowledge directly, incorporating the knowledge into another system for further action. Alternatively, results may be documenting and reporting to interested parties. This process also contains checking for and resolving potential conflicts with previously believed (or extracted) knowledge. As referred by several authors (e.g., [6]), the primary methodologies are inspired by KDD process as well as CRISP-DM.

The SAS Institute proposed a process of sampling, exploring, modifying, modeling, and assessing large volumes of data to discover previously unknown patterns. This process is called SEMMA, which is the acronym of sampling, exploring, modifying, modeling, and assessing. And it can be applied to business advantage. The SEMMA, data mining process, is appropriate for a variety of industries. It also provides methodologies for such diverse business problems as customer retention and attrition, house-holding, risk analysis, fraud detection, database marketing, market segmentation, affinity analysis, bankruptcy prediction, customer satisfaction, and portfolio analysis [3].

In 1996, four leaders of the nascent data mining market (Daimler-Benz, IntegralSolutions Ltd. (ISL), NCR, and OHRA) created CRISP-DM (cross-industry standard process for data mining). CRISP-DM is a complete data mining methodology and process model that provides anyone with a comprehensive blueprint for performing a data mining project. The CRISP-DM life cycle has six phases: (1) business understanding, (2) data understanding, (3) data preparation, (4) modeling, (5) evaluation, and (6) deployment [2]. Business understanding is the initial phase. It focuses on understanding the project objectives and requirements from a business perspective. The knowledge is converted into a data mining problem definition. It is also in this phase that the preliminary plan is designed to achieve the objectives. The data understanding phase starts with the initial data collection. Then, it proceeds

with activities to be familiar with the data: identifying data quality problems, discovering first insights into the data, or detecting interesting subsets to form hypotheses for hidden information. The data preparation phase covers all activities until the construction of the final dataset. At the end of this phase, the data ready is be used by modeling tool(s). Data preparation tasks are likely to be done multiple times and often not in any fixed order. This task includes table, record, and attribute selection as well as transformation and cleaning of data for modeling tools. In the modeling phase, various modeling techniques are selected and applied. And their parameters are calibrated to optimal values. Usually, there are several techniques for the same data mining problem type. Various techniques have specific requirements on the form of data. Thus, moving back to the data preparation phase is often necessary. As may be analyzed in several examples, e.g., [22], the selection of techniques is an iterative process. In the evaluation stage, the built project has a model (or models) that appears to have high quality. Before continuing to the final deployment of the model. it is also essential to assess the model and analyze the steps executed to build the model to be sure it appropriately reaches the business objectives more systematically. A crucial objective is to determine if there is some critical business issue that has not been sufficiently considered. A choice on the use of data mining results should be achieved at the end of this phase. Creation of the model is not typically the end of the project. Yet if the aim of the model is to increase knowledge of the data, the knowledge gained need to be organized and displayed in a way that the client can use it. However, conditional to the requirements, the deployment phase can be simple or complex. It may be just generating a report or implementing repeatable data mining process across the enterprise. Often, it is the client or another entity, not the data analyst, who carries out the deployment steps. However, even if the analyst does not carry out the deployment effort, the customer needs to understand upfront what actions need to be carried out in order actually to make use of the created models. On the other hand, using presentation techniques is critical in this phase [15, 25].

Analytics Solutions Unified Method (ASUM) is an iterative IBM SPSS Process to implement a data mining/predictive analytics project. It is based on an extended and refined CRISP-DM methodology. ASUM-DM has five phases: (1) analyze, (2) design, (3) configure and build, (4) deploy, and (5) operate and optimize. Nevertheless, the first three steps of ASUM (analyze, design, and configure and build) may be combined because data mining/predictive analytics projects are iterative by nature. It is also possible to add an optional project management process [4] (Fig. 2).

To deliver predictive analytics solutions and intelligent applications efficiently, Microsoft proposed the Team Data Science Process (TDSP). It is an agile, iterative data science methodology. TDSP also suggests how team roles work best together. The purpose is to improve team collaboration and learning [5]. TDSP includes best practices from Microsoft and other industry players to help toward successful implementation of data science projects. The purpose is helping companies realize the benefits of their analytics program. TDSP includes a suggested life cycle that may be used to structure your data science projects. The life cycle outlines the steps

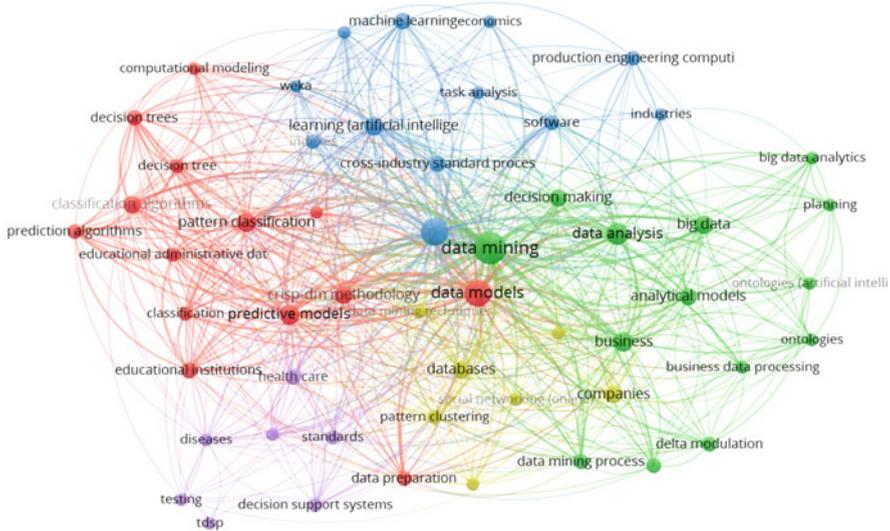


Fig. 2 Clusters of concepts related to data mining/data science methodologies

of a typical project. It is possible to adopt another data science life cycle, such as the CRISP-DM, KDD or organization’s custom process. This approach tries to be more comprehensive. But in fact, it is a patchwork of fussy “good practices”. Using as keywords the acronyms of the previous approaches (“SEMMA” OR “CRISP-DM” OR “ASUM” OR “TDSP”), we obtained a list of 126 papers from IEEE Xplore. Then, we used VOSviewer [30], in order to identify clusters of keywords. VOSviewer allowed the identification of five clusters. Some clusters were more focused on the methodologies, others on the techniques, and others on the strategies. Some clusters also were centered on the application field (Fig. 3).

This brief bibliometric analysis emphasized that it is important to consider other aspects complementary to the process. For example, the choice of techniques is a crucial dimension. Scheduling and budgeting may also be relevant to consider. In the approaches presented, the emphasis is in the process. Like any project, time is critical. Nowadays, ethics and privacy must also be considered [28]. It is also vital to consider the risk. And monitorization must be performed to conclude if the project went successfully [25]. But it is also essential to identify the possible roles involved and how do those roles may be organized. In this context, using techniques like RACI (responsible, accountable, consulted, informed) may contribute to improving project organization [17]. The way how information is analyzed may also help in the selection of the best way to success. This is why a criterion selection of tools should be followed. Many approaches may be used to select the most appropriate tools [19]. The following figure presents a flowchart developed by a group of researchers from MIT [14] representing the main decisions involved in the selection of the tools. This chart is at a high level, but it is essential in the decision processes of

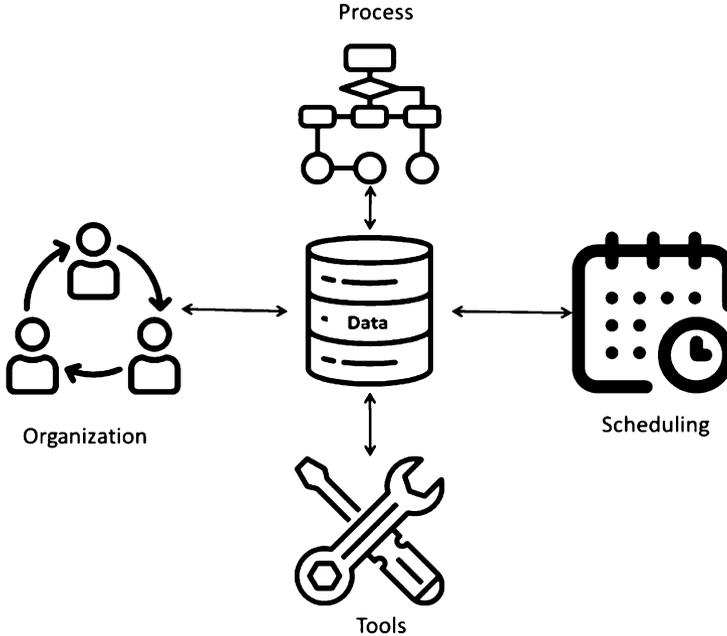


Fig. 3 POST-DS methodology

choosing Machine Learning tools. On the other hand, other techniques may be used in different phases. For example, in the first phase, it is vital to align corporate strategy and mission with the project mission. Some authors even suggest the use of data knowledge to select appropriate models in the context of the managerial decision [20].

Several techniques may also be used in order to select the most appropriate charts in order to create the best interfaces [15, 25]. Summarizing, the approaches presented for data mining, machine learning, and data science may be interrelated. CRISP-DM is one of the most used and also the one that inspired many of the major approaches. Nevertheless, other features may be added to this approach.

3 Proposing a Data Science Model

Following the literature review, it is possible to identify process, organization, scheduling, and tools as the four main blocks to tackle a problem of data science. The following figure represents the proposed model graphically:

Process: The POST-DS (Process Organization and Scheduling electing Tools for Data Science) describes the sequence of activities performed in a data science project. The list of the possible task is very well documented in CRISP-DM as

it was described previously. The first phase of business understanding is a phase where project scope, cost, and time are defined. Risk, ethics, and privacy must be considered at the beginning of the process but must also be managed throughout the project.

Schedule, Budgeting, and Scope: In project management, scheduling is an essential element. Each activity must be performed at a specific time. Often, if a project does not conclude in the expected, it is entirely useless. Other dimensions must also be considered, like budgeting and scope.

Organization: It is normal to have different roles in a data science project. The data engineer, data scientist, business analyst, or computer engineer are just possible roles in a data science project. Several people may perform these roles. It is essential to identify what are the roles in a specific project and how they participate in each phase or task of a project. Using a responsibility matrix, RACI is a possible solution. All those building blocks interact to extract knowledge from data.

Tools: Tools include not only the software used but also conceptual tools and resources. What are the sources of data? Open data is a solution or data must be bought. The software and experts are also essential resources. Open source is widely used in the context of data science. Often, free and open-source software is even better than proprietary software. But typically, there's a need for experts. In order to implement this approach, we proposed a template [29].

4 Using the Model

We implemented this approach in the context of a real situation. The purpose was to understand the spread of COVID-19. This project was not supported by any organization, corresponding to the combination of interests of several researchers and practitioners. Figure 4 shows a preliminary version of the implementation of the proposed approach. Some of the activities changed as well as the timings. And some of the roles also regrouped according to the skills of the persons involved in the project. The usage of acronyms is the following for roles: BA business analyst, DE data engineer, DS data scientist, WD web designer, and RACI responsible, accountable, consulted, informed.

In the first phase, a business analyst defined the business objectives, scope of the project, time, and budget, assessed the situation, and determined the data science goals. Then, a data engineer is typically responsible for data collection and quality assurance of this data. We obtained data from several sources on the Internet, like the European Union [26] and the United Nations [27]. A preliminary analysis allows identifying many gaps in the information. Data understanding and data preparation may be supported by using several charts. It is useful to understand the behavior of the variables available. Figure 5 shows an example of the number of cases per million in several countries. This preliminary analysis may be useful to identify possible incongruencies. For example, the reduced number of cases in China and the reduced number of recovered cases in Portugal are two examples that are at least

	BA	DE	DS	WD	Risk	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10	w11	w12	w13	w14	Tools and Resource
1 Business Understanding																				
1.1. Define Business Objectives																				
1.2. Identify ethical values and privacy	A/R				L															meeting
1.3. Assess Situation	A/R				L															meeting
1.4. Define Data Science Goals	A/R				L															meeting
1.5. Produce Project Plan	A/R	R	R		L															WBS, GANTT
2 Data Understanding																				
2.1. Collect Initial Data		A/R			H															open data, scraping,
2.2. Describe Data		A/R			L															use Jupyter/python/Pandas
2.3. Explore Data		A/R			M															use Jupyter/python/Pandas
2.4. Verify Data Quality			A/R		H															use Jupyter/python/Pandas
3 Data Preparation			A/R																	
3.1. Select Data			A/R		M															Meeting
3.2. Clean Data			A/R		M															use Jupyter/python/Pandas
3.3. Construct Data			A/R		M															use Jupyter/python/Pandas
3.4. Integrate Data			A/R		H															use Jupyter/python/Pandas
3.4. Format Data			A/R		H															use Jupyter/python/Pandas
4 Modeling																				
4.1. Select Modeling Techniques	I		A/R		H															MIT flowchart
4.2. Generate Test Design	I		A/R		H															use Jupyter/python/Pandas
4.3. Build Model	I		A/R		M															use Jupyter/python/Pandas
4.4. Assess Model	I		A/R		H															use Jupyter/python/Pandas
5 Evaluation																				
5.1. Evaluate Results, including ethical	A/R		R		H															use Jupyter/python/Pandas
5.2. Review Process	A/R				L															meeting
5.3. Determine Next Steps	A/R				L															meeting
6 Deployment																				
6.1. Plan Deployment	A		R	R	H															PowerBI or Flash
6.2. Plan Monitoring and Maintenance	A				M															meeting
6.3. Produce Final Report	A/R	R	R	R	M															PowerBI or Flash
6.4. Review Project	A/R		R		M															meeting

Fig. 4 POST-DS Example usage

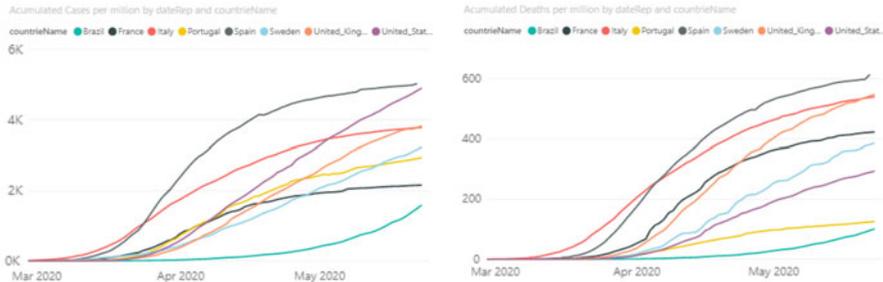


Fig. 5 Comparing the evolution of cases and death per million

suspicious. To solve some of those limitations, we try to cross several sources. But as a consequence of the nature of data, health data is sensitive data; it was challenging to verify the quality of data.

Data preparation and modeling are the responsibility of the data scientist. The business analyst evaluated the models with the support of the data scientist. In this context, the team used several tools. Those two phrases, data preparation and modeling, are often related. Often it is even necessary to collect more data. In this context, an epidemic curve was adjusted. Figure 6 shows a preliminary phase in the

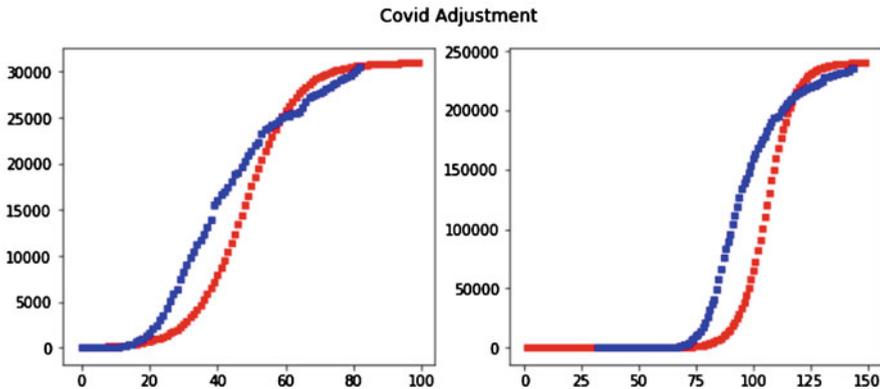


Fig. 6 Model adjustment

process of identification of the best model and adjustment of the parameters. But results from all the countries were very different.

A possible approach was identifying clusters of countries, according to the government strategy related to COVID. For example, the United States, Brazil, and Sweden showed a similar evolution.

This situation is different from other countries. Portuguese government implemented early measures. The United Kingdom shifted the approach. In this context, it is possible to estimate epidemic curves for each cluster. But the reality is much more complex, and GDP, temperature, humidity, and percentage of the aged population are also variables that may be used to identify other possible reasons not only of the spread of the disease but also the number of deaths.

Designer under the supervision of the business analyst deployed the model, incorporating its results in a website. In this context, several solutions were analyzed. A possible alternative consists of using a Microsoft solution. A possible solution consists of using Power BI. Another alternative consists in using Django or Flask and embed all the results. The price of the solution and expertise are two dimensions that must be evaluated. In fact, it is important to obtain results, but it is also essential to maintain results. The management of the data science project, in this case, gains from the assistance of such a methodology. In fact, it allows integrating the specified components, contributing to the more effective and efficient management of all the resources allocated to the project. Because it allowed the adjustment of expectations, clarifying the scope of the project, costs, and time, it also makes clear the tasks assigned to each person.

5 Conclusions

An overview of the evolution of data mining process models and methodologies is studied. Given that the methodologies analyzed were not complete, a new approach (POST-DS) was proposed. This approach includes the following components: processes, organization, scheduling, and tools. This approach is inspired particularly in the cross-industry standard process for data mining, but it intends to give additional guidelines. This methodology was applied in a specific data science project. The project was related to COVID-19. The application made it possible to conclude that this POST-DS can contribute to a better alignment of overall project management.

Acknowledgments We gratefully acknowledge the financial support from FCT (Fundação para a Ciencia e Tecnologia (Portugal)), national funding through research grant UIDB/04521/2020.

References

1. S. Aparicio, J. Aparicio, C. Carlos, Data science and AI: Trends analysis, pp. 1–6, in *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)* (2019). <https://doi.org/10.23919/CISTI.2019.8760820>
2. C. Shearer, The CRISP-DM model: The new blueprint for data mining. *J. Data Warehous.* **5**(4), 13–22 (2000)
3. SAS Institute Inc. SAS® Enterprise Miner™ 14.3: Reference Help. Cary, NC: SAS Institute Inc. (2017)
4. IBM Analytics solutions unified method-Implementations with Agile principles. 2016. Retrieved 15 Dez 2019. <https://doi.org/ftp://ftp.software.ibm.com/software/data/sw-library/services/ASUM.pdf>
5. G. Ericson, Z. Martens, K. Sharkey, C. Gronlund, Team Data Science Process Documentation. Retrieved 12 Jan 2020, from <https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/>
6. G. Mariscal, O. Marban, C. Fernandez, A survey of data mining and knowledge discovery process models and methodologies. *Knowl. Eng. Rev.* **25**(2), 137–166 (2010)
7. R.J. Brachman, T. Anand, The process of knowledge discovery in databases. *Advances in Knowledge Discovery and Data Mining*. American Association for Artificial Intelligence (1996), pp. 37–57
8. Two Crows Corporation Introduction to Data Mining and Knowledge Discovery, 2nd edn. Two Crows Corporation (1998). ISBN: 892095-00-0
9. Two Crows Corporation Introduction to Data Mining and Knowledge Discovery, 3rd edn. Two Crows Corporation (1999). ISBN 1-892095-02-5
10. S. Anand, A. Büchner, *Decision Support through Data Mining* (FT Pitman Publishers, London, 1998)
11. S. Anand, A. Patrick, J. Hughes, D. Bell, A data mining methodology for cross sales. *Knowl. based Syst. J.* **10**(7), (1998) 449–461
12. A.G. Buchner, M.D. Mulvenna, S.S. Anand, J.G. Hughes, An Internet-enabled Knowledge Discovery Process (1999), pp. 13–27
13. P. Chapman, et al., CRISP-DM 1.0: Step-by-step data mining guide. *SPSS inc* **9**, 13 (2000)
14. MIT What do you want the machine learning system to do? Intractive infographic (2019). <https://s3-us-west-2.amazonaws.com/getsmartergraphics/Courses/MIT+ML/M2/MIT+ML+M2+interactive+infographic.html>

15. C.J. Costa, M. Aparício, Supporting the decision on dashboard design charts, in *Proceedings of 254th The IIER International Conference* (2019), pp. 10–15
16. K. Adamiecki, “Harmonograf”. *Przegląd Organizacji* (1931) (translation to English)
17. A. Tewari, S. Mishra, S. Siddiqui, P. Upadhyay, Performance measurement at the requirement phase of software development life cycle, in *2nd International Conference on Computing for Sustainable Global Development (INDIACom)* (2015), pp. 1090–1094
18. J. Cabanis-Brewin, J.S. Pennypacker, Aligning projects to corporate strategy - Strategic Performance. Paper presented at PMI® Global Congress 2006—North America, Seattle (Project Management Institute, Newtown Square, 2006)
19. S. Bibi, I. Stamelos, Selecting the appropriate machine learning techniques for the prediction of software development costs, in *IFIP International Conference on Artificial Intelligence Applications and Innovations* (2006), pp. 533–540
20. S. Banerjee, A. Basu, A knowledge based framework for selecting management science models, in *Twenty-Third Annual Hawaii International Conference on System Sciences*, vol. 3, pp. 484–493 (1990). <https://doi.org/10.1109/HICSS.1990.20538>
21. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From data mining to knowledge discovery in databases. *AI Mag.* **17**(3), 37–37 (1996)
22. N. Fernandes, S. Moro, C. Costa, M. Aparício, Factors influencing charter flight departure delay. *Res. Transp. Bus. Manag.* 100413 (2019). <https://doi.org/10.1016/j.rtbm.2019.100413>
23. C. Costa, M. Aparicio, Analysis of e-learning processes, in *Proceedings of the 2011 Workshop on Open Source and Design of Communication*, New York, pp. 37–40 (2011). <https://doi.org/10.1145/2016716.2016726>
24. H. Fayol, *Administration industrielle et générale*, Dunod. (1916)
25. M. Aparicio, C. Costa, Data visualization. *Commun. Des. Q. Rev.* **3**(1), 7–11 (2015). <https://doi.org/10.1145/2721882.2721883>
26. <https://opendata.ecdc.europa.eu/covid19/casedistribution/csv>
27. <https://data.un.org/>
28. M. Piteira, M. Aparicio, C. Costa, Ethics of artificial intelligence: Challenges, in *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, Coimbra, pp. 1–6 (2019). <https://doi.org/10.23919/CISTI.2019.8760826>
29. C. Costa, J. Aparicio, POST-DS: A methodology to boost data science, in *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, Sevilla (2020)
30. N.J. Van Eck, L. Waltman, Text mining and visualization using VOSviewer. *ISSI Newsl* **7**(3), 50–54 (2011)

Shallow SqueezeNext Architecture Implementation on BlueBox2.0



Jayan Kant Duggal and Mohamed El-Sharkawy

1 Introduction

CNNs/DNNs are the core of the machine learning nowadays. ADAS or self-driving cars make use of them as the background computation algorithms. DNNs overcame the limitations of the traditional algorithms [1–4] which occupied more memory and are computationally expensive. DNN model performance represents model accuracy, model memory size, and model speed. Due to more intensive DSE of CNNs, macro CNN architectures were introduced earlier for ADAS or real-time embedded systems [1–4] such as SqueezeNet and SqueezeNext baseline architectures. Usually, DNNs are now trained and tested on the datasets such as CIFAR-10, CIFAR-100, ImageNet, MNIST, etc. The proposed Shallow SqueezeNext architecture is implemented on CIFAR-10 and CIFAR-100 [5] datasets, trained and tested on a GPU initially, and later was tested on a real-time embedded platform, BlueBox2.0 [6] by NXP (Fig. 1).

J. K. Duggal (✉)

Department of Electrical and Computer Engineering, IoT Collaboratory, Indianapolis, IN, USA
e-mail: jaydugga@iupui.edu

M. El-Sharkawy

Purdue School of Engineering and Technology, IUPUI, Indianapolis, IN, USA
e-mail: melshark@iupui.edu; stucker@iupui.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_8

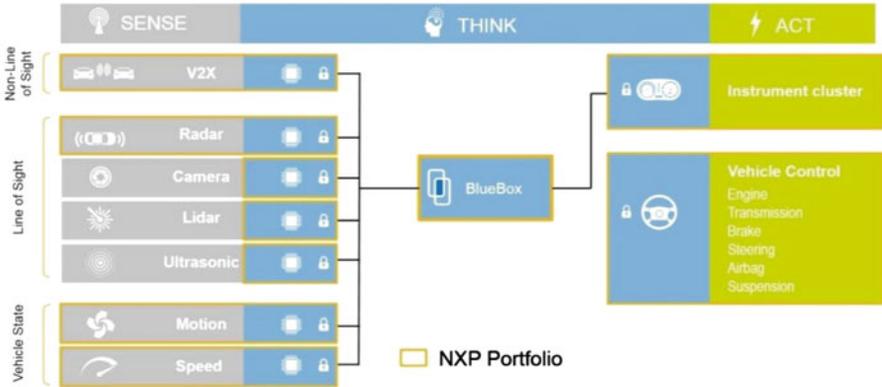


Fig. 1 NXP ADAS real-time network of sensors, BLBX2-DB Bluebox: development platform for self-driving vehicles. (<https://blog.nxp.com/automotive/sense-think-and-act-on-level-3-autonomous-drive-capabilities-today>)

2 BlueBox2.0 Development Platform

BlueBox2.0 by NXP [6] is a real-time development platform that provides the required performance, functional safety, and automotive reliability to develop self-driving cars. It is an ASIL-B and ASIL-D compliant hardware system, an integrated package for creating autonomous applications such as ADAS systems and driver assistance systems. It is comprised of three independent systems on a chip that are S32V234, vision processor; LS2084A, computing processor; and S32R274, radar microcontroller.

It uses one of the Cortex-A72 layer cape processors out of the 8 processors and an embedded vision chip S32V234. It includes Level 1, deploying collision warnings and automatic brakes and maintaining a set vehicle distance from others; Level 2, technology implementation of car steering, brake, and acceleration automatically within limited conditions and constraints, not eliminating the need of a human driver; and Level 3, autonomous applications such as transferring the complete hand over safety-critical functions in certain situations from the driver. The challenge here is providing autonomous cars the ability with more computation and memory resources with a fail-proof system. It operates on the independent embedded Linux OS BSP package for both the S32V and LS2 processors with the help of RTMaps. It functions as the central computing unit of the system, therefore providing the ADAS system to be capable of deploying efficient and better CNN/DNN models (Fig. 2).

2.1 Vision Processor (S32V234)

The S32V234 microprocessing unit [6] offers an ISP, powerful 3D GPU, dual APEX-2 vision accelerators, automotive-grade reliability, functional safety, and, for

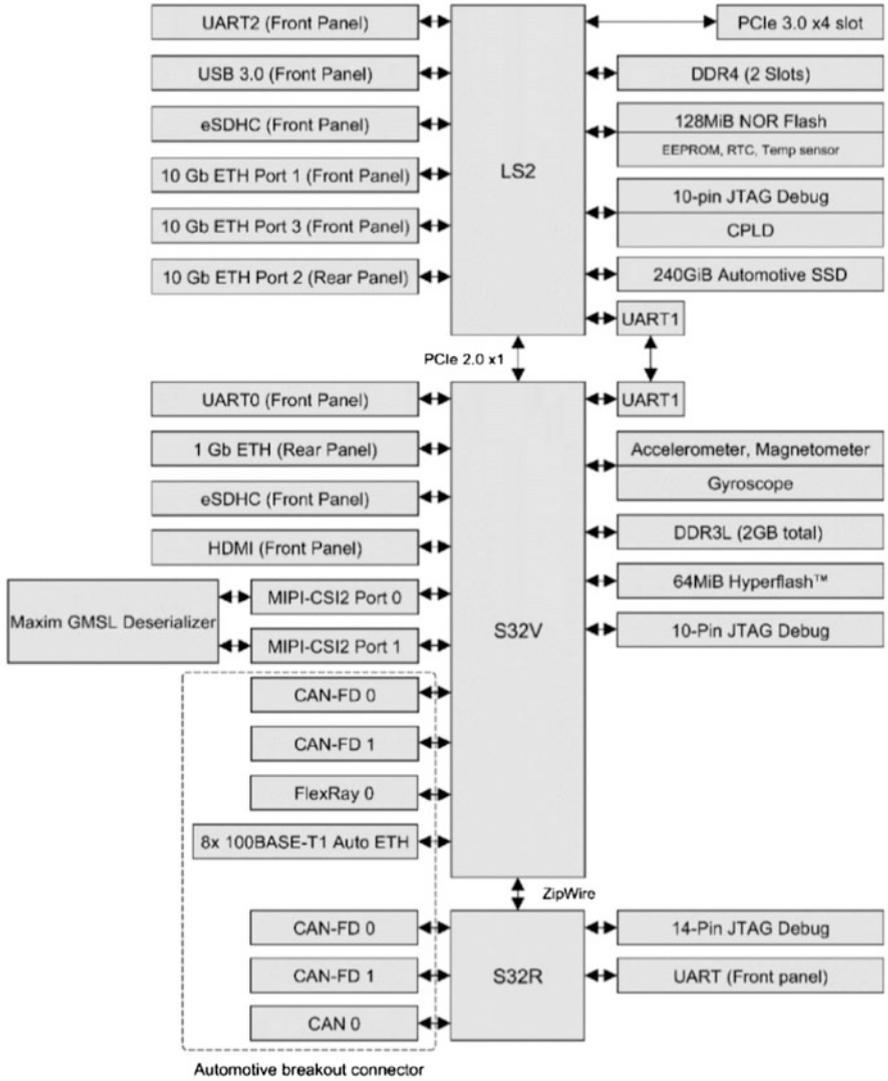


Fig. 2 Hardware architecture for BlueBox 2.0 by NXP. (<https://www.nxp.com/design/development-boards/automotive-development-platforms/nxp-bluebox-autonomous-driving-development-platform:BLBX>)

supporting computation-intensive ADAS, NCAP front camera, object detection and recognition, surround view, automotive and industrial image processing, and also ML and sensor fusion applications. It is a second-generation vision processor family and member of the 32-bit ArmR©CortexR©-A53 S32V processors and is supported by S32 Design Studio IDE for the development of vision applications. Design studio includes a compiler, debugger, Vision SDK, Linux BSP, and graph tools.

It is a vision-based processor designed for computationally intensive applications related to CV applications. The processor comprises of ISP available on all MIPI-CSI camera inputs, providing the functionality to integrate multiple cameras. It contains APEX-2 vision accelerators, a GPU designed to accelerate CV function, four ARM Cortex-A53 cores, and an ARM M4 core designed for embedded related applications. The processor can operate on Linux BSP, Ubuntu 16.04 LTS, and NXP vision SDK. The processor boots up from the SD card. The SD card is interfaced at the front panel of the BlueBox2.0.

2.2 LS-2084A

The LS2 processor [6] in BlueBox2.0 is a high-performance computing processor platform. It consists of eight ARM Cortex-A72 cores and 10 Gb Ethernet ports, supports a high total capacity of DDR4 memory, and features a PCIe expansion slot. It is also a convenient platform to develop the ARMV8 code and is connected to a Lite-On Automotive SSD via SATA, to provide a large memory size for software installation. It also consists of an SD card interface that allows the processor to run Linux BSP and Ubuntu 16.04 LTS as an OS platform on the BlueBox platform.

In this research, the software enablement on the LS2084A and S32V234 SoC is deployed using the Linux BSP. The LS2084A and S32V234 SoC are installed with Ubuntu 16.04 LTS which is a complete, developer-supported system. It contains the complete kernel source code, compilers, and toolchains, with ROS kinetic and Docker package. The QorIQ LS2 family of processors delivers unmatched performance and integration for smarter and capable CNN/DNN networks. The eight-core QorIQ LS2084A and the four-core LS2044A multicore processors offer Arm Cortex-A72 cores with advanced, high-performance data path and network peripheral interfaces required for networking, datacom, wireless infrastructure, military, and aerospace applications. The data path architecture combined with a software toolkit provides a higher level of hardware abstraction and makes software development fast and simple.

2.3 Real-Time Multisensors Applications (RTMaps)

RTMaps [6] is an asynchronous high-performance platform and has the advantage of an efficient and easy-to-use framework for fast and robust development. It is a modular toolkit for multimodal applications. The easiest way to develop, test,

validate, benchmark, and execute applications is designed for the development of multimodal-based applications, thus providing the feature of incorporating multiple sensors such as camera, LiDAR, and radar. It has been tested for processing and fusing the data streams in the real-time or even in the post-processing scenarios. It consists of several independent modules that can be used in different scenarios.

RTMaps runtime engine is an easily deployable, multi-threaded, highly optimized module. It is designed in a context to be integrated with third-party applications and accountable for all base services such as component registration, buffer management, time stamping threading, and priorities.

RTMaps component library consists of the software module which can be easily interfaced with the automotive and other related sensors and packages such as Python, PyTorch, TensorFlow, C++, MATLAB Simulink models, and 3-D viewers, etc., responsible for the development of an ADAS application.

RTMaps studio is the graphical modeling environment with the functionality of programming using Python packages. The development interface is available for Windows- and Ubuntu-based platforms. Applications are developed by using the modules and packages available from the RTMaps component library.

RTMaps embedded is a framework which comprises of the component library and the runtime engine with the capability of running on an embedded x86 or ARM capable platform such as NXP BlueBox, Raspberry Pi, DSpace MicroAutoBox, etc. RTMaps embedded v4.5.3 platform is tested with NXP BlueBox. It is used independently on the Bluebox2.0. The RTMaps remote studio operating on a computer provides the graphical interface for the development and testing purpose. The connection between the computer running RTMaps remote studio and the embedded platform can be accessed via a static TCP/IP as shown in Fig. 3.

3 Shallow SqueezeNext Architecture

Shallow SqueezeNext architecture [7, 8] is derived from SqueezeNext [9], SqueezeNet [10], and MobileNet [11] architectures. It is a shallower architecture version of the high-performance SqueezeNext [8, 12], another architecture developed during this research.

Key strategies for Shallow SqueezeNext architecture:

- Using bottleneck modules, resolution, and width multipliers to manage the depth and width of the proposed architecture.
- Using only in-place operations in all layers except in the layers where we have a gradient change operation. Carefully sandwiching the ELU in-place layer in the basic block of the proposed architecture between a convolutional layer and a batch normalization layer.
- An element-wise addition skip connection [9] is implemented. This allows it to have a flexible architecture without the vanishing gradient problem.
- Adding a dropout layer [13] at the end of four-stage configuration after the average pooling layer.

Fig. 3 RTMaps connection

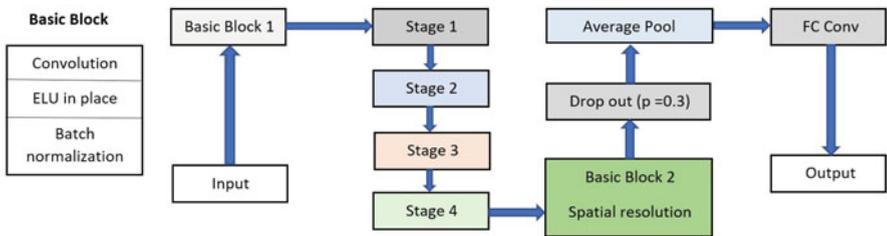
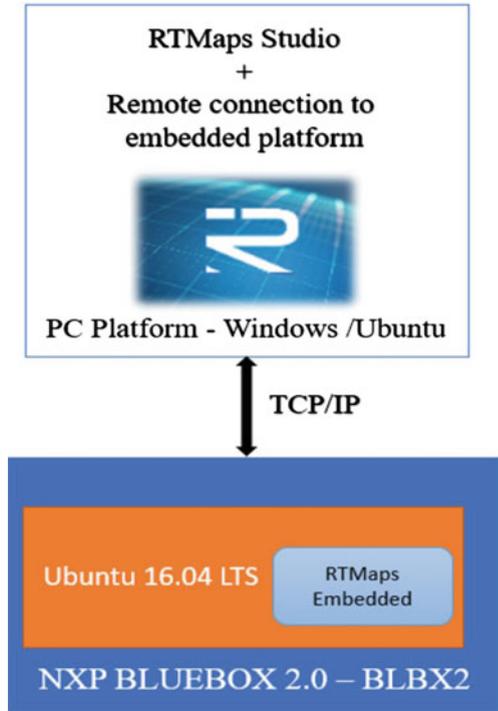


Fig. 4 Illustration of Basic Block (left) and Shallow SqueezeNext architecture with (1,2,8,1) four-stage configuration (right)

- Minimizing the use of max pooling layers or any kind of pooling layer is used to reduce the loss and quality of the image features (Fig. 4).

Bottleneck is the backbone of this architecture shown in Fig. 5 (left) along with four-stage configuration illustrating the bottleneck block arrangement in the (1,2,8,1) four-stage configuration with individually marked stages of the architecture.

4 Shallow SqueezeNext Deployment

Shallow SqueezeNext architecture is trained on GPU RTX 2080ti first, and then testing is done on BlueBox2.0 using RTMaps studio. RTMaps provide support for PyTorch. Figure 6 represents the process of deploying the proposed Shallow SqueezeNext architecture classifier on the BlueBox2.0 development platform. RTMaps studio initiates a connection to the execution engine using TCP/IP which runs the software on BSP Linux OS installed on BlueBox2.0. RTMaps provides a python block to create and deploy PyTorch code.

Python code for RTMaps comprises of three function definitions: birth(), core(), and death(). The **birth ()** module is executed one time for setting up and initializing the python environment. **Core()** is an infinite loop function to keep the code running continuously. **Death()** is used to perform cleanups and memory release after the python code terminates. Due to this, ease and flexibility within RTMaps and organized structure make it easy for developing a modular code. For maximum utilization of available eight ARM Cortex-A72 cores to run the RTMaps embedded, the LS2084A processor is used. The debug functionality and graphical interface

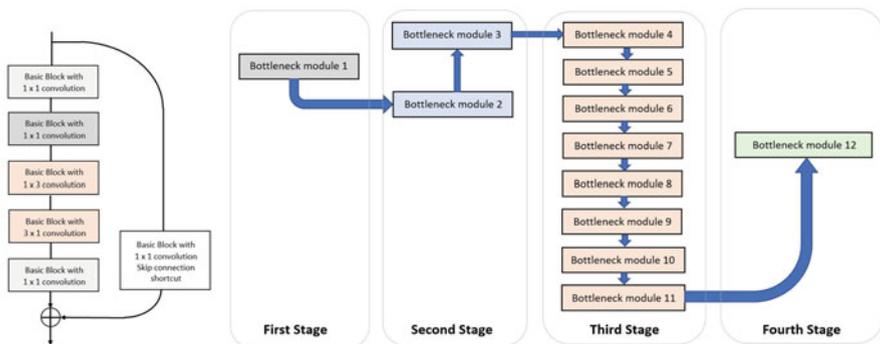


Fig. 5 (left) Illustration of Shallow SqueezeNext’s bottleneck module, (right) illustration of four-stage (1,2,8,1) configuration of the Shallow SqueezeNext architecture

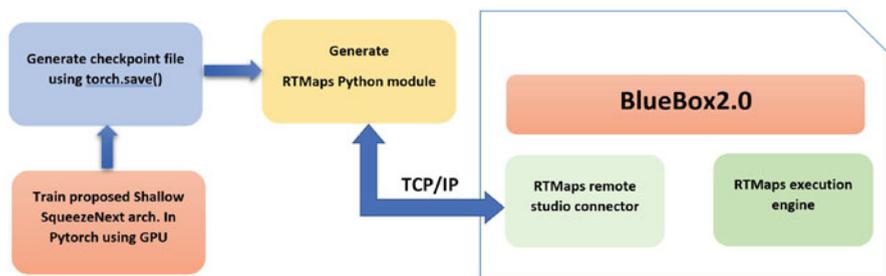


Fig. 6 Illustration of Shallow SqueezeNext BlueBox2.0 implementation [29]

the RTMaps embedded is connected with the RTMaps Studio in the desktop PC, illustrated in Fig. 3 using the remote engine connectivity provided over TCP/IP.

5 Results

5.1 *Shallow SqueezeNext Results*

Shallow SqueezeNext achieved a better reduced model size (0.115MB) from the SqueezeNext baseline model (PyTorch implementation) size (9.525MB). It had achieved very small model sizes along with better model speeds when used in resource-starved systems, and without any need for compression. The format of the models in the tables described below, for instance, Shallow SqueezeNext-14-1.5x-v1, depicts 14-layered resolution architecture with 1.5-width multiplier, and version 1. The Shallow SqueezeNext architecture is considered to be implemented in memory-constraint systems or environments.

Table 1 represents the experiments done with different models of Shallow SqueezeNext implemented with various four-stage configuration layers, width, and resolution multiplier. The insight gained from this table is that SGD and AdaBound optimizers perform better than their counterparts. Shallow SqueezeNext-06-0.5x-v1 model implemented with SGD optimizer had achieved the second-best model size, 1.1 MB, with minimum model speed, 8 s consumed per epoch for computation (least is better as this will attain less model time) with an accuracy of 59.69%. This model is implemented with Xavier initialization and SGD momentum with the Nesterov optimizer. Here, SGD is preferred over AdaBound as with it the Shallow SqueezeNext-14-1.0x-v1 model attains a better accuracy of 62.12% but with a model size of 6.90 MB at a model speed of 20 s per epoch due to the high number of four-stage configuration or resolution multiplier. Shallow SqueezeNext-14-1.0x.v1 model, if implemented with less number of four-stage configurations or resolution

Table 1 Shallow SqueezeNext results with CIFAR-100

Model	Optimizer	Accuracy%	Model size(MB)	Model speed (sec)
Shallow SqueezeNext-14-1.0x-v1	AdaBound	62.12	6.90	20
Shallow SqueezeNext-09-0.5x-v1	Adam	58.27	1.4	11
Shallow SqueezeNext-09-0.5x-v1	Adamax	33.73	1.4	13
Shallow SqueezeNext-09-0.5x-v1	Adagrad	35.57	1.0	9
Shallow SqueezeNext-09-0.5x-v1	AdaBound	51.84	1.4	12
Shallow SqueezeNext-09-0.5x-v1	Adadelta	44.40	1.4	12
Shallow SqueezeNext-09-0.5x-v1	ASGD	57.39	1.0	9
Shallow SqueezeNext-09-0.5x-v1	RMSprop	27.73	1.4	11
Shallow SqueezeNext-09-0.5x-v1	Rprop	16.84	1.4	25
Shallow SqueezeNext-06-0.5x-v1	SGD	59.69	1.1	8

multiplier, will not attain a decent accuracy in comparison to the other models. The other models with the least model size are not chosen as their accuracies are lower than the preferred Shallow SqueezeNext-06-0.5x-v1 with SGD and had more model speed (least is better).

In Table 2, Shallow SqueezeNext-06-0.4x-v1 model with a model size of 115 KB comprising of (1,1,1,1) four-stage configuration and 0.4x-width multiplier is 10x smaller than the SqueezeNext-23-1x-v1 2.586 MB and approximately 11x smaller than SqueezeNet v1.0 3.013 MB (both baseline models shown in Table 3). This table illustrates the trade-off for model accuracy and model size. To attain better model size, model accuracy is sacrificed and along with model speed. All models in this research are trained from scratch on the datasets that are CIFAR-10 and CIFAR-100 [14] without the use of the transfer learning method. Shallow SqueezeNext-21-0.2x-v1 illustrates the effect of the width multiplier. If we reduce width multiplier from 0.575x to 0.2x, 0.4x, and 0.5x, it will achieve a better model size but sacrificing some model speed. If we study linearity of relationships here between model size, speed, and accuracy, this will give an insight that the relationships here are nonlinear. Table 2 represents, in general, some of the better results with model accuracy, speed, and size trade-off. Shallow SqueezeNext-06-0.4x-v1 model with 0.2717 MB or 272 KB is considered to be the best least sized model. Instead of Shallow SqueezeNext-06-0.125x-v1 model that had achieved the least model size of 115 KB, the model accuracy suffered too much in comparison to Shallow SqueezeNext-06-0.4-v1. Other models had little more model size with better accuracies. From Table 2, Shallow SqueezeNext-09-0.5x-v1 and Shallow SqueezeNext-06-0.575x-v1 were implemented on BlueBox2.0 by NXP due to better model accuracy with decent model speed.

Table 3 compares the proposed Shallow SqueezeNext architecture model results with SqueezeNet and SqueezeNext baseline architectures. Different model variations of the Shallow SqueezeNext architecture here represent DNN models with better model accuracy, model speed (decent model speed +in comparison to

Table 2 Shallow SqueezeNext results with CIFAR-10

Model	Width, Resolution	Accuracy%	Model size(MB)	Model speed (sec)
Shallow SqueezeNext-06-0.125x-v1	0.125x, 1111	66.40	0.115	7
Shallow SqueezeNext-14-1.5x-v1	1.5x, 1281	91.41	8.72	22
Shallow SqueezeNext-21-0.2x-v1	0.2x, 22141	90.27	1.814	27
Shallow SqueezeNext-06-0.575x-v1	0.575x, 1111	81.80	0.449	6
Shallow SqueezeNext-06-0.4x-v1	0.4x, 1111	81.97	0.2717	9
Shallow SqueezeNext-09-0.5x-v1	0.5x, 1141	87.73	0.531	11

Table 3 Result comparison with SqueezeNet and SqueezeNext trained from scratch on CIFAR-10

Model	Accuracy%	Model size (MB)	Model speed (sec)
SqueezeNet-v1.0	79.59	3.013	4
SqueezeNet-v1.1	77.55	2.961	4
SqueezeNext-23-1x-v1	87.15	2.586	19
SqueezeNext-23-1x-v5	87.96	2.586	19
SqueezeNext-23-2x-v1	90.48	9.525	22
SqueezeNext-23-2x-v5	90.48	9.525	28
Shallow SqueezeNext-06-0.575x-v1	81.50	0.449	6
Shallow SqueezeNext-06-1.0-v1	82.86	1.24	8
Shallow SqueezeNext-09-0.5x-v1	87.30	0.531	11
Shallow SqueezeNext-14-1.5x-v1	91.41	8.72	22
Shallow SqueezeNext-21-0.2x-v1	90.27	1.814	27

*All results are three average runs with SGD; LR is 0.1

Table 4 Shallow SqueezeNext results deployed on Bluebox2.0, trained and tested on CIFAR-10 dataset

Model	Accuracy%	Model size (MB)	Model speed (secs)
Shallow SqueezeNext-14-1.5x-v1	90.50	8.72	22
Squeezed CNN	79.3	12.9	11
Shallow SqueezeNext-06-0.575x-v1	81.50	0.449	6
Shallow SqueezeNext-09-0.5x-v1	87.30	0.531	11

SqueezeNet baseline), and model size. Shallow SqueezeNext-09-0.5x-v1 model is 6x times smaller than SqueezeNet-v1.0 and 5x times smaller than SqueezeNext-23-1x-v1. The macro architectures (SqueezeNet and SqueezeNext) are already improved CNN/DNN architectures than the traditional CNN/DNN architectures.

5.2 BlueBox2.0 Implementation Results

BlueBox2.0 platform deploys the GPU trained image classifier, and then the classifier is tested on the platform. The network parameters for each model are saved and loaded from a checkpoint file for testing on BlueBox2.0 using PyTorch save and load method.

Table 4 compared the Shallow SqueezeNext result with Squeezed CNN architecture [15] which confirms that the Shallow SqueezeNext architecture is better and efficient in terms of model performance. Shallow SqueezeNext-14-1.5x-v1 model attained a better model accuracy of 90.50% than all other models mentioned in Table 4 and had a better model size and model speed in comparison to Squeezed CNN architecture (SqueezeNet-based architecture implemented previously on BlueBox2.0). Here, the preferred model of choice is the Shallow SqueezeNext-09-0.5x-v1 model as attained a model size of 0.531 MB, 16x smaller than the

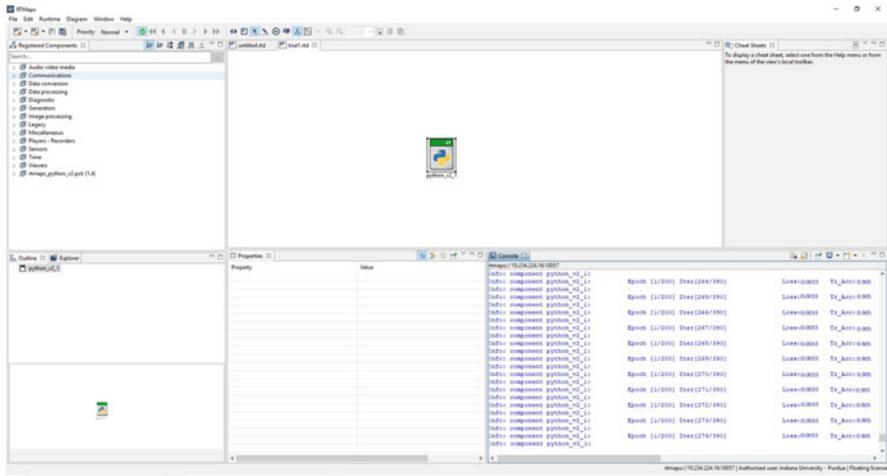


Fig. 7 Shallow SqueezeNext result on BlueBox2.0

Shallow SqueezeNext-14-1.5x-v1 and 24x times smaller than Squeezed CNN with a decent model speed of 11 s per epoch. The Shallow SqueezeNext architecture was supposed to be implemented or deployed in a memory-constraint system or environment. It can be observed here that the accuracy of Shallow SqueezeNext-14-1.5x-v1 with (1,2,8,1) four-stage configuration deployed on BlueBox2.0 in Table 4 when compared to Table 2 Shallow SqueezeNext-14-1.5x-v1 with exactly same model result (model trained and tested on RTX GPU) dropped a little when deployed on the real-time platform, Bluebox2.0. Figure 7 illustrates the deployment result of Shallow SqueezeNext testing on BlueBox2.0 by NXP, attaining a model accuracy of 90.5% using a python block implemented within RTMaps.

6 Conclusion

In this paper, based on the insights [1–6, 13, 16–20] from the existing DNNs, fine hyperparameter tuning, and architecture modifications, the Shallow SqueezeNext architecture is proposed and further implemented on a real-time platform, Blue-Box2.0 by NXP. Shallow SqueezeNext architecture has 120x fewer parameters than AlexNet and had a model size less than 0.5MB. It is 6x times smaller than SqueezeNet. With Shallow SqueezeNext architecture, DNN models achieved a small model size (0.531MB), model accuracy (90.50%), and model speed (6 s per epoch). The abovementioned various Shallow SqueezeNext models were successfully deployed on Bluebox2.0. All the Shallow SqueezeNext model results obtained decent model accuracies but majorly achieved a really small model size without quantization. Therefore, almost all the Shallow SqueezeNext architecture-based models can be successfully deployed on a real-time platform with limited resources

or within a resource-starved environment. In the research, Shallow SqueezeNext was trained and tested from scratch on both CIFAR-10 and CIFAR-100 datasets without any transfer learning to have a fair comparison with other architectures.

References

1. A. Shah et al., Deep residual networks with exponential linear unit. arXiv preprint arXiv:1604.04112 (2016)
2. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
3. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 1–9
4. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision. arXiv preprint arXiv:1512.00567 (2015)
5. K.O. Stanley, R. Miikkulainen, Evolving neural networks through augmenting topologies. *Evol. Comput.* **10**(2), 99–127 (2002)
6. S. Venkitachalam, S.K. Manghat, A.S. Gaikwad, N. Ravi, S.B.S. Bhamidi, M. El-Sharkawy, Realtime Applications with RTMaps and Bluebox 2.0. in *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, pp. 137–140. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (World-Comp) (2018)
7. J.K. Duggal, Design Space Exploration of DNNs for Autonomous Systems. Dissertation Purdue University Graduate School (2019)
8. J.K. Duggal, M. El-Sharkawy, Shallow SqueezeNext: An Efficient & Shallow DNN, in *2019 IEEE International Conference of Vehicular Electronics and Safety (ICVES)* (IEEE, 2019)
9. A. Gholami, K. Kwon, B. Wu, Z. Tai, X. Yue, P. Jin, S. Zhao, K. Keutzer, SqueezeNext: Hardware-Aware Neural Network Design. arXiv preprint arXiv: 1803.10615 (2018)
10. F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5MB model size. arXiv preprint arXiv:1602.07360 (2016)
11. A.G. Howard, et al., Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
12. J.K. Duggal, M. El-Sharkawy, High Performance SqueezeNext for CIFAR-10, in *2019 IEEE National Aerospace and Electronics Conference (NAECON)* (IEEE, 2019)
13. N. Srivastava et al., Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)
14. A. Krizhevsky, V. Nair, G. Hinton, Cifar-10 (Canadian institute for advanced research). <http://www.cs.toronto.edu/kriz/cifar.html> (2010)
15. D. Pathak, M. El-Sharkawy, Architecturally Compressed CNN: An Embedded Realtime Classifier (NXP Bluebox2. 0 with RTMaps), in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)* (IEEE, 2019), pp. 0331–0336
16. S. Chetlur, et al., cudnn: Efficient primitives for deep learning. arXiv preprint arXiv:1410.0759 (2014)
17. S. Ruder, An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747 (2016)

18. L. Luo, Y. Xiong, Y. Liu, X. Sun, Adaptive gradient methods with dynamic bound of learning rate. arXiv preprint arXiv:1902.09843 (2019)
19. T.B. Ludermir, A. Yamazaki, C. Zanchettin, An optimization methodology for neural network weights and architectures. *IEEE Trans. Neural Netw.* **17**(6), 1452–1459 (2006)
20. D. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289 (2015)

Dark Data: Managing Cybersecurity Challenges and Generating Benefits



Haydar Teymurlouei and Lethia Jackson

1 Background

Around 90% of all big data can be classified as dark data [1]. According to IBM, the high percentage of data gathered from sensors and analog-to-digital conversions never gets used which is considered dark data. This category of data is difficult to understand, analyze, and not used in any decision-making. Additionally, the governance and oversight of data risk assessments for about 69% of a company's stored data have absolutely no value to the organization [2]. Most enterprising information technology (IT) teams fight to identify and manage the dark data such as files, documents, emails, and instant messages which can be found behind every corporate firewall within file shares as well as cloud-based applications involving numerous tools. The risks and a clear picture of a company's security picture depend on the type and quality of the collection of dark data that is available for further investigation. A recent survey suggests that "fifty-four percent" of data in organizations is stale, and organizations have over one thousand stale sensitive files [3].

2 Introduction

The term dark data is used to describe the category of dormant and unmanaged content. Light data are the opposite of dark data. Light data, categorized and managed content, are used to promote consistency of filing, support record

H. Teymurlouei (✉) · L. Jackson
Department of Technology & Security, Bowie State University, Bowie, MD, USA
e-mail: hteymurlouei@bowiestate.edu

management, and facilitate access, storage, protection, and retention. Dark data are unused or unidentified content that sits outside of the normalcy of data and its use [4]. Professionals in IT need to become aware of abandoned information that may cause hidden risks which lie at the edge of their enterprise platforms [5]. The critical aspect of how data are categorized and managed is the platform in which organizations rely upon to meet compliance obligations and ensure decisions are made with accurate and relevant information that is not actionable with dark data. Rising at unprecedented rates, dark data have caused IT and business professionals to work collaboratively to generate a strategy that can assist with managing the governance, oversight, and risk of the uncategorized content sitting at the edges of the organization's enterprise [5]. The type of dark data that can appear in most companies depends on the industry and the type of information collected or processed. To identify and classify dark data may require searching the depths of storage environments with various tools and application solutions because manually manipulating dark data is difficult. The following are common information examples of dark data [6].

- Account information
- Analytics reports/survey data
- Biometric data
- The call records of the customer
- Emails and attachments
- Location data
- Human resource information
- Inactive databases, unused customer/system/machine information
- Internet of Things (IoT) diagnostics and status updates
- Log files
- The hidden files from application

Figure 1 provides details on how the dark data can be divided into different areas. Organizations face challenges identifying the examples of dark data listed above. For this reason, they are considered hiding in the dark.

3 Collection of Dark Data

Data are viewed as currency and/or exchange and/or marketplace. The process of transforming data for collection, conversion, and visualization defines an organization's data pipeline to analyze how their industry is performing [8]. Even through the data pipeline process, unstructured data sources are often overlooked because of the size, arrangement, or unreadability of the data. The explosion and volume of data has caused the use of automated tools and collection processes to streamline new data sources used to create additional tools [8]. The understanding, management, and conversion of the dark data presents the need for an action plan to accurately

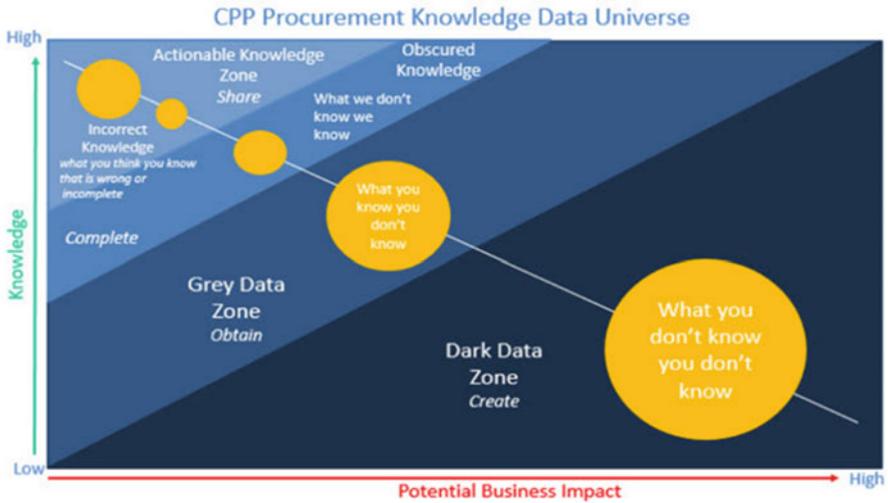


Fig. 1 Dark data zones [7]

make industry-based decisions that are meaningful and provide deeper analyses that minimize risks while revealing value for all data.

Ongoing management, collection, control, and adaptation of dark data can be made efficient with technology. A data governance platform as a management tool would prove useful to information managers, compliance and legal officers, and business users bringing a consistent data lifecycle management to all data including dark data. This governance platform should provide flexibility in a secure cloud environment that will allow on-demand scaling, lock down capabilities for sensitive information, organization for discovery, recovery, updates, disposal, and reuse, align with legacy architecture, and work across multiple data sources to include real-time data.

3.1 Dark and Unstructured Data

Dark data, known as unstructured data, are harder to analyze than light data [2]. Due to the volume of big data, some of the data are not distributed through the data pipeline for various reasons made by the organization. Data become dark when organizations do not realize the business value of the data. Data that are not utilized require significant storage space. The IBM mentioned that “80% of big data is considered to be dark and unstructured data” [9]. A small number of businesses used audio, image video files, and machine and sensor-generated information by the Internet of Things to explore and expand the view of dark data [9]. Nevertheless, light data can also be a part of dark data. A large portion of

data is never processed, which means that some portion of the data is structured, analyzed, contained, and utilized. As an illustration, two structured datasets (light datasets) can maintain in their individual storage facility. These datasets could have dissimilar formats, be stored in different systems, operated under different teams, and considered redundant, obsolete, and trivial [2]. Combining these datasets could provide additional insights if explored in the depths of the deep web.

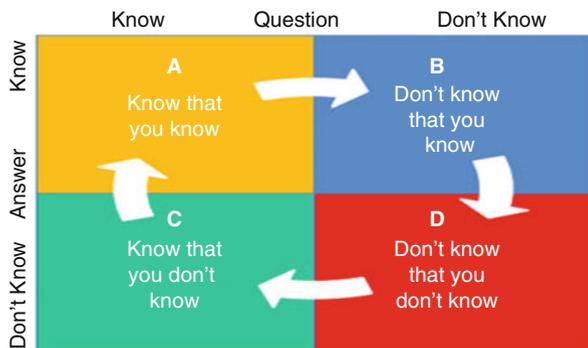
3.2 Dark Data as a Treasure

Both dark and light data are valuable to a business assets and worth. Dark data, on the other hand due to their unorganized structure, hold important information that is not available in any other format [2]. Organizations continue to pay to collect and store dark data to maintain compliance and to have the ability to explore the data in the future. Some of the organizations use manual extraction to interpret and explain the data. Other organizations use deep learning technologies and tools to extract dark data at a much faster and cheaper rate than manual extraction [2]. Figure 2 offers the details of the dark data that can be divided into different areas: the data that are known and the other data that are unknown without being secured.

4 Dark Data Risks

According to IBM, 80% of big data could be classified as dark data and consumes the same costs of light data but provides no real value [9]. The need for expanded storage and maintenance costs coupled with the risk of security are consequences of the large amount of dark data [10]. Users can easily access light data from a structured database based on the category, type of information, and parameters which make light data more readily available. Dark data are inaccessible and all mixed together with all different types and categories of data making it very difficult

Fig. 2 Known and unknown data [7]



to extract. Hackers find all data quite beneficial to cause disruption, which is the reason why organizations purpose to eliminate dark data until moving through the data pipeline, as it receives the same treatment as light data [10]. To provide the kinds of data that most organizations gather, those risks might embrace some or all the following.

- *Legal and regulatory risk.* Data protected by governance, mandate, or regulation may expose confidential and financial information that is located within dark data collections which may lead to legal and financial accountability [11].
- *Intelligence risk.* Dark data could cause compromise of a business performance, as it may include proprietary or sensitive information that may preclude the business from operating in sound judgment, maintaining the competitive edge and/or adversely affecting the business's bottom line [11].
- *Reputation risk.* Dark data could cause a data or other kinds of breaches [11].
- *Opportunity costs.* Businesses that decide not to invest in mining dark data may miss out on opportunities to grow the business, learn more about employees and customers, decrease costs, and increase productivity and profits, avoiding liabilities [11].

4.1 Data Challenges with Dark Data

As mentioned earlier, there are various problems associated with dark data such as containing sensitive and/or proprietary information and security risks that can become more prevalent as time goes by. As dark data continue to be produced, it is a waste of storage capacity that could be used to store useful assets and information. Additional storage means extra overhead costs, a predominate anxiety for organizations that consume big data [12].

4.2 Turn Dark Data into Active Revenues

There are several methods that can transform dark data to active. Organizations should allow collaboration between data science, marketing, and business intelligence together to create opportunities to innovate ways that can activate dark data [13]. Prior to investigation, in some instances dark data appear dismal and tedious. On the contrary, using investigative methods can turn dark data into rich insight for a better business. The following are a few examples of the benefit of dark data.

- *Login.* When traveling to various cities/countries, user login information contains their Internet protocol (IP) addresses which could lead to the person's locations. Based on the information of their whereabouts, a virtual map could be created of their travel pattern? [13].

- *Additional guest names on a reservation.* The reservation that contains more information as it concerns additional guest names gives insight into a user's social network. The social network provides rich insight into the social network graph of the user to include family and workplace. Everyone included on this social network can be accessed to obtain additional information about each of them to include age, gender, and behavioral traits [13]. Additionally, a recursive cycle exists for each person located on the social network, which can be extended into their network as well.
- *Mobile phone data.* Mobil phone data are very rich for marketplace, advertising and marketing opportunities which serve as a spring board for innovative ideas for new products and marketing effectiveness. From a business standpoint, the organizations can also know their customer's shopping and purchasing habits in real time [13].

4.3 Make Dark Data to Work

The most crucial point of understanding dark data is that they do not have to remain dark [14]. Companies have recognized that they have the potential to turn dark data into actionable and interesting insights for the company's gain such as networking machine data, customer support logs, and legacy system logs [14].

5 Recommendations

Dark data are redundant, frequently forgotten-about data that a user collects [15]. It is recommended that dark and unstructured data be analyzed and monitored. Real-time monitoring of data will give businesses the power to forecast future threats and eliminate current ones. Analyzing the data with the use of technologies will allow for businesses to utilize the data in various ways beneficial to the company.

5.1 Analyzing

To begin the process of destroying unnecessary dark and unstructured data, you must first search the physical storage devices such as hard drives and compact disc read-only memory (CD-ROMS) for the data. Next, use tools to speed up and streamline the process to wipe, analyze, and convert the data into light data by identifying, classifying, and separating the data. Some of the data may not have any value but, during this process, you may find information that could improve the business. There are some common ways to wipe storage devices clean of unwanted data: physical, encryption, and overwriting. Deleting files of unwanted data can be accomplished

through data sanitization, a preferred process to ensure that data do not fall into criminal hands. Data sanitization deliberately, eternally, and irreversibly removes or destroys the data stored on a memory device. Once the data have been sanitized, they cannot be reversed and there is no opportunity to retrieve the data even after using advanced forensic methods [10].

5.2 Benefits of Analyzing Dark Data

The arrival of new kinds of data, such as network dark data and digital photography, delivers analytics on new predictive modeling and real-time analysis using automated processes [8]. Mining dark networking data for network security performance and network activity patterns allows organizations to check for stress points and optimize resource use. Records of customer support conversations and other customer data points are often underutilized but can provide an opportunity for dark data analytics [2]. Digital photography involves the use of digital cameras which follow a file-naming convention of numbering picture files sequentially and organize downloaded images in a date-based file organization. Photographs embed metadata which organize and sort images as well as contain information on who made it, when, and using what. Manually entered metadata usually describe the image itself keywords, notes, and copyright information. To search for photographs using an exact location, person, or event must be done manually because there is no correlation between the photograph formation date and the context of what is being searched. To solve the manual search issue, smart applications using machine and deep learning tools can scan through document contents, find documents based on keywords, and utilize photo-organizing tools to identify faces, landmarks, and features to classify photographs [2].

5.3 Cybersecurity Monitoring

Monitoring the dark web helps to notify if an incident occurred. Incident response is essential to react without delay. Procedures to maintain and analyze security logs and events in real time can be used to assist in monitoring security-related incidents, which have proven to be necessary to quickly detect appropriate countermeasures. Real-time evaluation becomes more important as attackers and hackers become more sophisticated. Organizations under cyberattack cannot afford to be inefficient which will only increase damage and losses from lack of processes to protect data. Without active and real-time monitoring and analysis of security logs, valuable information will likely be compromised.

5.4 Analytics Technologies

Implementing analytics technologies to help reveal insights not only within raw data already in possession but also in derived data represents a powerful business opportunity. Yet, dialing up the data mining and analysis efforts while importing large stores of dark data from external sources can lead to questions about data accuracy, integrity, legality, and appropriateness of use. These are questions that few organizations can afford to ignore today. On the flip side, a deep analysis of more data from a variety of sources may also yield signals that could potentially boost cyber and risk management efforts. Indeed, the dark analytics trend is not just about deploying increasingly powerful analytics tools against untapped data sources. From a cyber-risk perspective, this trend is about manipulating these and other tools to inspect both the data in your possession and third-party data purchase.

6 Methodology

Security is always at the forefront of any organization's IT strategy. Usually, this strategy focuses on the perimeter of the network to prevent unauthorized access or attacks from malicious parties that are not associated with the organization. There might be chances that criminals execute their malicious activities, stealing dark data and placing organizations at risk. On the other hand, if the collected data are considered valuable and utilized, organizations will automatically strengthen security procedures. Such a step will allow organizations to safeguard their digital assets against data theft.

Figure 3 shows data collected in large amounts containing technical information. Some of these collected data may be considered valuable, light data, and some of the collected data may not be considered valuable, dark data. Light data are accessible, can easily be found and placed into a clear and structured format that contains categories, and differentiated using various parameters. Dark data have different types of mixed data making it hard to find specific information. Figure 3 indicates that dark data usually are a blockage between raw data and structured data. According to IBM, only 20% of the data is likely to be analyzed by management. In the far-left block, many devices are represented (personal computer (PC), printer, switch, router, laptop, etc.). Eighty percent of the data from these devices is considered to be invaluable information. However, these data are actually network activity from the users with valuable data like files, email, and confidential data. By identifying these dark data, it could be beneficial to control potential cybersecurity threats.

The amount of data that has been collected from an organization is indicated in Fig. 4 on the left side. Insider threats can be identified in the same data. These data possess significant value to the businesses and the cybersecurity threats which are related to the given areas. There are particular dark data that are providing



Fig. 3 Management area for structure data



Fig. 4 Information in the dark data

the company heads up if any local machine has been bug-in for future threats. On the other side, if there is a threat in a specific system that has been hidden, it will assist to identify any backdoor communication with hackers. The management only looks at small (10%) percentage of the data and a high percentage (80%) is the dark data according to IBM, as Fig. 4 shows the values from the company. Example log files could provide clues to the website visitor behavior. Footage from a surveillance video or recordings of customer calls could provide unstructured customer sentiment data. Geolocation data can offer new insight into your shipping and logistic operations [16]. Whatever the view may be, the key is that it is new and untapped information. Dark data analysis encourages the company to take another look at these data to see if there is any additional value or insight to gain. Dark data

are valuable because they often hold information that is not available in any other format. Therefore, organizations continue to pay the cost of collecting and storing dark data for compliance purposes and with hopes of exploiting the data in the future. Because of this value, organizations sometimes resort to human resources to manually extract and annotate the data, and then enter them into a relational database, even though this process is expensive, slow, and error prone. Deep learning technologies perform dark data extraction faster and with much better accuracy than human beings. Dark data extraction is less expensive and uses less engineering effort when using these techniques and tools.

6.1 Dark Data User Activities

Monitoring data especially “Dark Data” in an organization can ensure individuals who are supposed to have access can track logged-in activities and prevent a breach as well as manage passwords efficiently. To overcome cybersecurity challenges, log files keep track of events that are produced from hardware and software operations. Information retrieved from log files can range in the notification from informational events to warnings and then to critical errors. The event log can alert and raise a red flag to an administrator that a significant file has been modified, deleted or that there has been an unsuccessful attempt.

6.1.1 Using Security Log as an Auditing Tool

Security Logs may be used as an auditing tool to track actions performed within a system. Some of the following Security Log IDs may be used to track various actions that may be considered malicious.

- 5140: A network share object was accessed.
- 4660: An object was deleted.
- 4624: An account was successfully logged on.
- 4663: An attempt was made to access an object.
- 4656: A handle to an object was requested.
- 4659: A handle to an object was requested with intent to delete.

Most organizations have IT management. The management might experience and have two different data. Dark data can contain structured data which are not analyzed. Usually, the high percentage of data tends to be unstructured and low percentage to be structured.

Figure 5 illustrates dark data in an 8-year period from 2011 to 2019. This demonstrates how valuable dark data are. The above graph specifies how much dark data have been collected. The counter displays extensive numbers of data files that have gone dark; nonetheless, the majority have not been analyzed. In this case, an example of the logged data is the event log file. This file indicates network

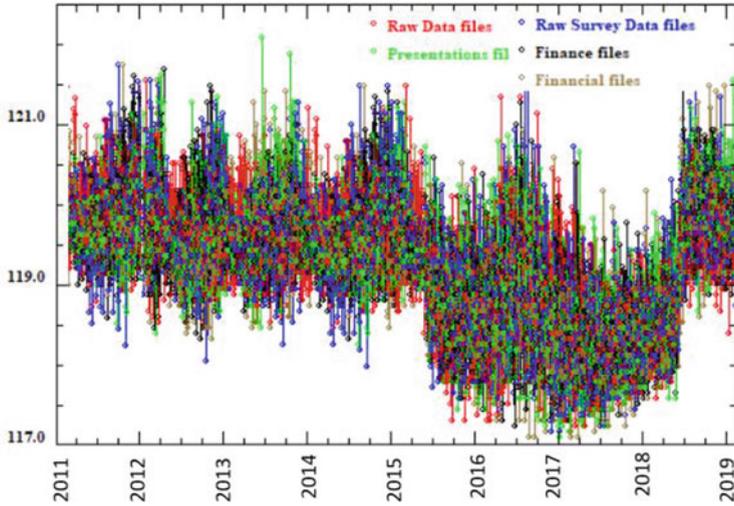


Fig. 5 Sample information in the dark data

activity in the system. By analyzing this file, there is potential to forecast future cyber threats. This event is beneficial for financial aspects and security reasons. Moreover, it describes the date and time of the event ID number when it was deleted or accessed. This predicts the threats and it will assist to identify the future of cyber threats. This method detects data in the system that is stored in locations without permission.

Figure 6 shows a monthly section that displays raw survey data files only. These data are valuable to the organization because they provide more detailed and specific data. It is possible to see time and date.

Another example of a detailed cross-section of the dark data is presentation files, which are gathered in the organization. This is presented in Fig. 7 over a pattern of the month. Such examples of dark data in Figs. 6 and 8 must be effectively processed. It is important to define the logging requirements and design and implement the logging architecture. This will provide all events captured in one place and format. Now, it is time to put the log-monitoring process into action. An effective log monitoring is not a tool or a technology, although rather than a process that requires continuous improvement. The key activities associated with that process are described in Fig. 8.

7 Conclusion

The more data organizations collect and store, the more they will have to take care of the data’s security. As dark data are considered as useless data by most

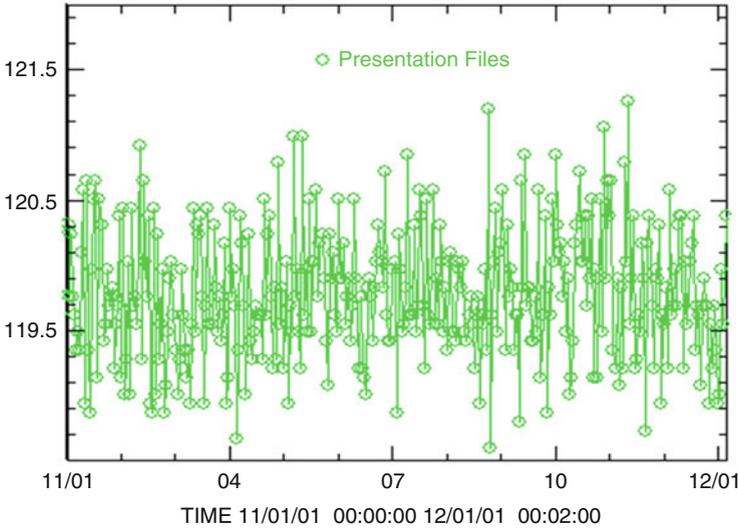


Fig. 6 Plotted 1 month number of raw survey data files

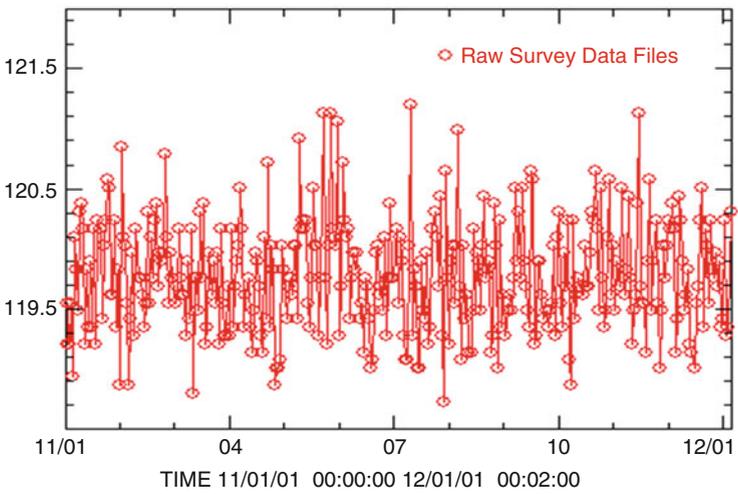


Fig. 7 Plotted 1 month number of presentation files

organizations, there are great chances that organizations fail to enforce strict security controls. The data that may seem unimportant to organizations may be interesting to hackers. Dark data certainly represent unused opportunities that many companies are letting go of because of process, investment, and technology constraints. In a sense, this failure to use dark data also makes big data collection, which is an

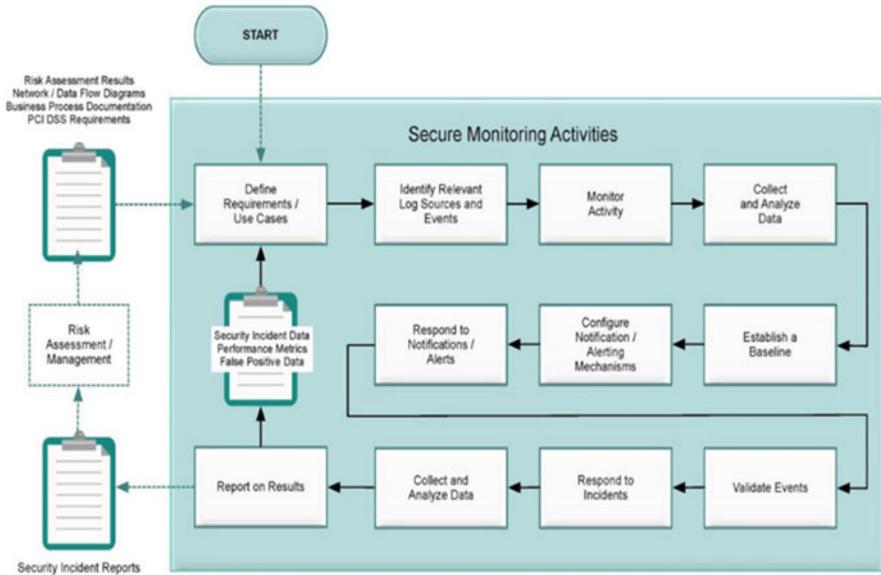


Fig. 8 Monitoring process [17]

outstanding exercise, a partial failure. Though the investments needed to tap dark data potential may be costly, the effort is worth the investment.

References

1. Analyzing “Dark Data” for Traders Using IBM Watson and Bluemix. (2017), Retrieved February 7, 2020, from Altoros website: <https://www.altoros.com/blog/analyzing-dark-data-for-traders-using-ibm-watson-bluemix/>
2. From data to knowledge: Dark data. (2018), Retrieved February 7, 2020, from [IBM.com](https://www.ibm.com/developerworks/library/ba-data-becomes-knowledge-3/index.html) website: <https://www.ibm.com/developerworks/library/ba-data-becomes-knowledge-3/index.html>
3. From data to knowledge: Dark data. (2018), Retrieved from [IBM.com](https://www.ibm.com/developerworks/library/ba-data-becomes-knowledge-3/index.html) website: <https://www.ibm.com/developerworks/library/ba-data-becomes-knowledge-3/index.html>
4. N. Anderson, Control Dark Data in the New Age of Compliance and Security. (n.d.), Retrieved from https://www.ciosummits.com/Control_Dark_Data_in_the_New_Age_of_Compliance_and_Security.pdf
5. Dark Data, Big Data, and Your Data: Creating an aCtion Plan for Information governanCe. (n.d.). Retrieved from https://www.ciosummits.com/Dark_Data_Big_Data_Your_Data_Creating_an_Action_Plan_for_Information_Governance.pdf
6. Dark Data – The Blind Spots in Your Analytics | iDashboards Blog, (2020), Retrieved February 7, 2020, from [Idashboards.com](https://www.idashboards.com/blog/2019/01/30/dark-data-the-blind-spots-in-your-analytics/) website: <https://www.idashboards.com/blog/2019/01/30/dark-data-the-blind-spots-in-your-analytics/>
7. T. Dingeldein, 4 Dark Data Mining Insights for Small Business Owners. (2018), Retrieved February 7, 2020, from [Capterra.com](https://blog.capterra.com/dark-data-mining/) website: <https://blog.capterra.com/dark-data-mining/>

8. Hortonworks Big Data Maturity Model HORTONWORKS BIG DATA MATURITY MODEL. (2016), Retrieved from <http://hortonworks.com/wp-content/uploads/2016/04/Hortonworks-Big-Data-Maturity-Assessment.pdf>
9. J. Colchester, Dark Data – Systems Innovation. (2018), Retrieved February 7, 2020, from Systems Innovation website: <https://systemsinnovation.io/dark-data/>
10. Destroying Dark and Unstructured Data | IDM Magazine. (2018), Retrieved February 7, 2020, from Idm.net.au website: <https://idm.net.au/article/0012194-destroying-dark-and-unstructured-data>
11. Figure 2f from: Irimia R, Gottschling M (2016) Taxonomic revision of *Rochefortia* Sw. (Ehretiaceae, Boraginales). *Biodivers. Data J.* 4: e7720. <https://doi.org/10.3897/BDJ.4.e7720>. (n.d.). doi: <https://doi.org/10.3897/bdj.4.e7720.figure2f>
12. The risks of Dark Data for your organization | Migrato(2016), Retrieved February 7, 2020, from Migrato website: <https://www.migrato.nl/the-risks-of-dark-data-for-your-organization/>;
M. Tulloch, Shedding light on dark data: Is it an opportunity or risk? (2019) Retrieved February 7, 2020, from TechGenix website: <http://techgenix.com/dark-data-risk/>
13. M. Ross-Smith, Dark Data: The hidden billion dollar opportunity – SmartData Collective. (2016), Retrieved February 7, 2020, from SmartData Collective website: <https://www.smartdatacollective.com/dark-data-hidden-billion-dollar-opportunity/>
14. bhuthesh@kenome.io, Kenome: An Enterprise Knowledge Graph Company. (2017), Retrieved February 7, 2020, from Kenome.io website: <https://www.kenome.io/>
15. K. Kirkham, Why Companies Need To Pay Attention To Their Dark Data – blog.100tb.com. (2018), Retrieved February 7, 2020, from blog.100tb.com website: <https://blog.100tb.com/why-companies-need-to-pay-attention-to-their-dark-data>
16. What is Dark Data and How Can You Use It? – UC Today. (2019), Retrieved February 7, 2020, from UC Today website: <https://www.uctoday.com/unified-communications/what-is-dark-data-and-how-can-you-use-it/>
17. Standard: PCI Data Security Standard (PCI DSS) Effective Daily Log Monitoring. (2016), Retrieved from <https://www.pcisecuritystandards.org/documents/Effective-Daily-Log-Monitoring-Guidance.pdf>

Implementing Modern Security Solutions for Challenges Faced by Businesses in the Internet of Things (IoT)



Haydar Teymurlouei and Daryl Stone

1 Background

Internet of Things (IoT) plays the role of an expert's technical tool, by allowing physical resources into smart entities, through present network infrastructures. Its prime emphasis is to provide smart and seamless services at the user end, deprived of any disruption. The IoT paradigm is intended to create a complex information system with the combination of sensor data acquisition, efficient data exchange through networking, machine learning, artificial intelligence, big data, and clouds [1]. On the other hand, gathering information and maintaining the confidentiality of an independent entity, running together with privacy and security provisions in IoT, is of vital importance. There are a lot of pervasive uses present in the human environment of things or objects, such as radio-frequency identification (RFID) tags, sensors, actuators, mobile phones, and smart embedded devices. These devices, through unique addressing schemes, can effectively communicate and interact with each other and work together to reach a common goal of making the system easier to operate and utilize. The objects that will be connected will be adaptive, intelligent, and receptive [2]. The IoT will be altogether a new environment in which the current Internet will be smartly supported, by a new range of smart embedded devices. An optimistic approach to individuals in accepting the unfolding changes transported by IoT will also support its growth. The uncertainty and business risk are always present in any new technology. In the case of IoT, it is observed that many of the dangers are not physically present or are somewhat distorted or misstated.

H. Teymurlouei (✉) · D. Stone

Department of Technology & Security, Bowie State University, Bowie, MD, USA

e-mail: hteymurlouei@bowiestate.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_10

105

2 Introduction

IoT is an evolution in the integrative world. Sensors and networking devices, microcontrollers and microprocessors, are some of the basic building blocks of the IoT and these are in pervasive use nowadays. These devices are more effective today, as they get smaller and more affordable to create. Size and affordability will allow IoT, in the long run, to include billions of interconnected devices over the internet. Looking at the development of IoT, it will include numerous developers from around the world, who will produce various product types. Currently, the major requirements, like hardware and software assets, are either available in less capacity or they are underdeveloped. This leads to the fact that security and confidentiality concerns of IoT devices were not properly addressed over the past decade.

2.1 *IoT Management*

Currently, the Internet of Things technology is providing the ability to connect each object to a network. The Internet of Things offers a chain of associated people, objects, applications, and data over the Internet for remote control, interactive, services integration, and management [3]. IoT management requirements have several significant differences when compared to a traditional network. Figure 1 shows how the IoT architecture can be managed [4]. Figure 1 displays that managing such devices presents a different problem, as compared to traditional Internet or telephony management [5]. Fortunately, the restrictions imposed on the IoT devices are well understood right now and have influenced the creation of evolving management approaches, less dependent on human interference. The IoT viability depends on the scalability, generality, and comprehensiveness of its management.

3 Challenges

IoT components are applied to utilize divergent protocols and technologies. As a result, these mechanisms have intricate configurations and underprivileged design. This makes the business owner face obstacles in selling the products to the consumers. The following are challenges that tend to be common on IoT. Figure 2 indicates the IoT challenges based on the percentage.

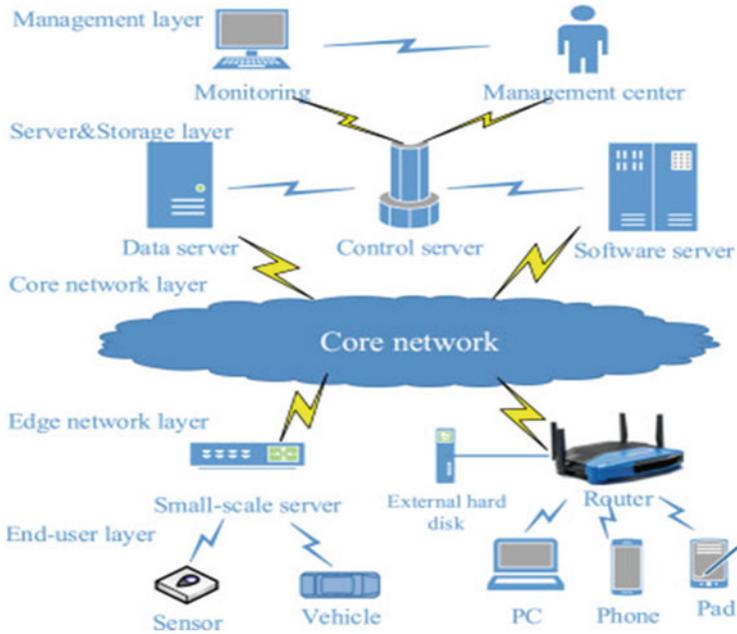


Fig. 1 Manageable IoT architecture [4]

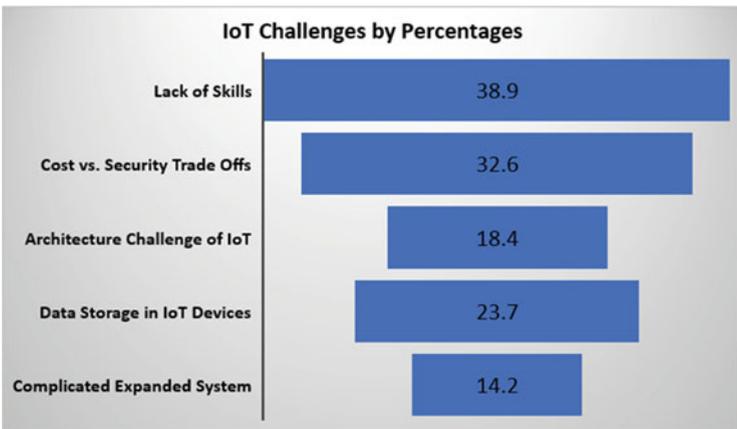


Fig. 2 IoT challenges

3.1 Privacy

Privacy is the main concern while designing and developing IoT devices, consequently, addressing these concerns must be a high priority. New technology usually

has scope for abuse, and it is smarter to solve the issue, before it influences privacy and innovation [6]. It is the responsibility of manufacturers, standards organizations, and policymakers to address all the possible threats to the product [6]. As a part of network layer security, manufacturers must deliberate about the implementation of new security protocols that will be key factors in guaranteeing the successful end-to-end transmission of delicate data [6].

3.2 Security

Security is one of the greatest critical matters in IoT development. Providing security for IoT technology is an enormously real challenge to overcome. Since the IoT technology has a wide scope, it is necessary to focus on the security challenges related to various aspects, including performance, work efficiency, costs, data, wireless sensor networks (WSNs), and other security challenges [6]. In addition, IoT has caused major security issues that have grabbed the attention of numerous public and private sector companies [7]. Including such an enormous number of new hubs to the system will provide attackers with a larger platform to invade [6]. Business owners must implement the malware to capture an infinite number of IoT gadgets that are being used in basic applications like smart-home devices and closed-circuit cameras.

3.3 Limitations

While IoT technology offers many advantages in several areas and solves a range of problems in different sectors, it still faces a range of different limitations. Particular skills and expertise are essential factors required for designing, implementing, developing, and managing security that must be considered [6]. The interruption of any of these factors may cause damage to the security system in the adoption of IoT technologies. There are limited infrastructure resources. IoT devices typically have little processing capabilities and limited memory [6]. It is a major challenge for IoT hardware manufacturers and software developers to design comprehensive security measures within a low memory of 64–640 KB [6]. Moreover, it is important to have enough space for security software to defend against security threats. As a result, the central processing unit (CPU) and memory are limited in IoT. Nevertheless, the IoT generates many innovations in versatile areas of life and it has significant benefits. It includes many limitations that traditional systems did not present [6]. As a result of the IoT rather complicated system, there are different limitations associated with IoT devices.

3.4 *Connectivity*

Another significant challenge of the IoT is to connect several devices. This communication tends to result in an attack on a currently existing structure and the technologies associated with it. Presently, a centralized client server architecture is being utilized to authenticate, authorize, and connect several terminals in a network [4]. While this model may be appropriate for the existing situation, it may not be scalable, to supply future needs where billions of devices will be part of a single network. The IoT scenario will turn the current centralized system into a bottleneck. Investments in cloud clusters of servers are needed to deal with the humongous quantity of information exchange and prevention of unavailable servers, which can lead to a total system shutdown [8].

3.5 *Requirements*

The essential security issues in IoT systems require protecting two critical aspects, which are confidential data and identity authentication [6]. Furthermore, the main requirements in information security are considered: data availability, data confidentiality, data integrity, and data authenticity. Any breach of these areas will cause security damages or problems to the IoT system [6]. Correspondingly, each of the layers of IoT must meet these requirements. Data availability is crucial in the IoT. It contributes to ensure that users have access to the security and reliability of available data. An IoT system needs to provide the backup of vital information to prevent data loss. Some attacks cause harm related to data availability [6]. Data confidentiality requires the protection of data using specific encryption techniques and mechanisms to prevent data disclosure and any unauthorized access to IoT equipment and devices. Data integrity refers to protecting valuable and sensitive information from the risk of cybercriminals [6]. Several things affect data integrity, for example, server downtime. The cyclic redundancy check (CRC) is a way to guarantee data integrity and detect message encryption errors by adding a fixed-length value to detect network errors in IoT [6]. Authentication and authorization issues play an essential role in IoT security. These verify the identity of users or devices and then grant access to nonsuspicious IoT objects or services [6].

3.6 *Data Storage*

Once the size amount of data rises at a very high rate, data storage becomes a major problem. Similarly, data storage affects data protection [6]. As soon as stored data are damaged, it is difficult to back up all the stored data. There are no specific criteria to ensure that data distributed within IoT devices are securely

transferred to the main data center. This is because the process of transferring data is not synchronized, making it disproportionate to the data center. This presents yet another major challenge for data storage and data management companies in terms of emerging tools and standards that address data and provide security properly [6].

3.7 Network

The IoT system has different devices that use different communication networking protocols. IoT networking protocols can be divided into smart device networks and traditional networks, which are used to increase data rates [9]. Smart networking protocols are expected to adopt the protocols already established in WSNs and Machine-to-Machine (M2M) communications. Building a networking protocol is not an uncomplicated task, as it should fulfill the requirements of ease-of-use, appropriate cost, and performance of the whole system [9]. Additionally, choosing a suitable network topology for the protocol is another issue. Nevertheless, the mesh topology is the most suitable choice for wireless communication in smart environments [9]. Hence, different communication protocols and different network topologies unveil a significant challenge that needs to be handled.

3.8 Maintenance

Maintenance is a serious challenge to acknowledge as high percentage of new devices are now inundating the internet. These devices may belong to different vendors who have already gone out of business, and their devices may be full of bugs that nobody will ever be able to solve. Moreover, numerous vendors do not adhere to the suggested practice of upgrading their devices to the latest platforms and security fixes. Not only can their devices create a big challenge in the overall IoT performance, it is considered as a weak point, which can be attacked easily and affect the whole IoT network.

4 Recommendations

Companies can take advantage of the following recommendations, to build a secure IoT environment. By using these recommendations, businesses will be able to overcome IoT obstacles and create a trust with their consumers.

4.1 Risk Assessment

It is important for companies who sell a smart-home device to secure each one by constructing a risk assessment. By generating a risk assessment, the company would accomplish multiple security aspects. It would manage an inventory of each device that is connected to their network improving their awareness. Next, the company would be able to prioritize attacks. Afterward, the users could begin applying their security measures more efficiently, instead of wasting valuable resources on an appliance that causes a loss of customers' personal information.

4.2 Monitoring

Monitoring devices such as temperature control or camera footage are managed poorly from both the user and service provider. For example, camera monitoring systems come with a manual that holds default password configurations and installation. After installing the camera, customers do not bother to change the default password, leaving an attacker who knows the default password an easy path inside. The companies which produce IoT products must advise their clients to monitor their devices. To prevent this, the user must follow the manual start to finish. In terms of temperature control systems, restricting physical access to them is a start. Additionally, applying an intrusion detection system (IDS) or virtual private network (VPN) to the Internet service provider (ISP) creates a secure connection to the outside network as well as an alarm system. Most IoT appliances industries do not consider security and focus more on marketing and sales. Applying security is often overlooked or very limited, since the devices are made to perform their tasks and nothing else. This means implementing scripts or software is difficult. To avoid the outsiders from discovering open ports and devices, the corporation must recommend securing a network by restricting what users can do on the network. This also applies to connected devices on unsecured connections. It is recommended to keep the devices on one central network, instead of having them on separate open connections. Since this will expose the private data the appliance collects to anyone connected to the unsecured access point.

4.3 Add Security

All IoT appliances must be added to the security layer to prevent future incidents to the customers. There is a multitude of steps to adding this security, but some addition utilizing subtraction done as well. Turning on Wi-Fi Protected Access 2 (WPA2) encryption with Advanced Encryption Standard (AES), Temporary Key Integrity Protocol (TKIP) is a great decision, nonetheless, AES is always preferable

[10]. WPA2 AES is currently the most secure encryption standard available, until WPA3 becomes widely available [10]. Disabling Wi-Fi Protected Setup (WPS) and Universal Plug and Play (UPnP) feature, just as the Bluetooth button allows open pairing with any device in its range, the WPS button allows the router to reconnect to the devices on its network which could be a potential vulnerability [10]. Disabling remote access prevents hackers from accessing the admin panel, even if somehow the hacker was able to wirelessly breach the panel [10]. While accessing the admin panel is not necessarily a problem, doing so wirelessly could present a problem, because the login credentials are sent and may be intercepted by hackers. Disabling remote access prevents hackers from retrieving the admin panel during wireless access and thus provides added security.

4.4 Cloud and Gateway Architectures

IoT appliance must contain a cloud architecture. Cloud architectures and security, which allow for the collaboration between devices like the monitor, collect, store, and process data from IoT devices [1]. Gateway architectures, which work on the same Local Area Network (LAN) with other IoT endpoints, can improve interconnection and interoperability between smart devices and act as the central management point that assigns the coordinates for IoT devices.

5 Methodologies

The Internet of Things is a widely used term for set of technologies. For businesses to deliver a secure great value, it is important to consider how to overcome IoT challenges in business. By implementing various methodologies, such as segmenting networks to secure hardware, and Virtual Local Area Networks (VLANs) and VPNs, it will allow companies to use IoT in a secure environment.

5.1 Segmenting Networks

Network segmentation is a simple thought that has been used by network administrators for many years, nevertheless lately there is a real need to apply this concept in the home [11]. Lots of people back up their laptops and desktops to network drives, which is a good start. Normally, the backup is happening routinely in the evening, when the laptop is not being used. At the same time, the other users are watching the news, while one of them is printing important documents and another is playing a video game. In the architecture of Fig. 3, all that traffic runs through the one home router. In the architecture below, all the backups and printing take place behind the

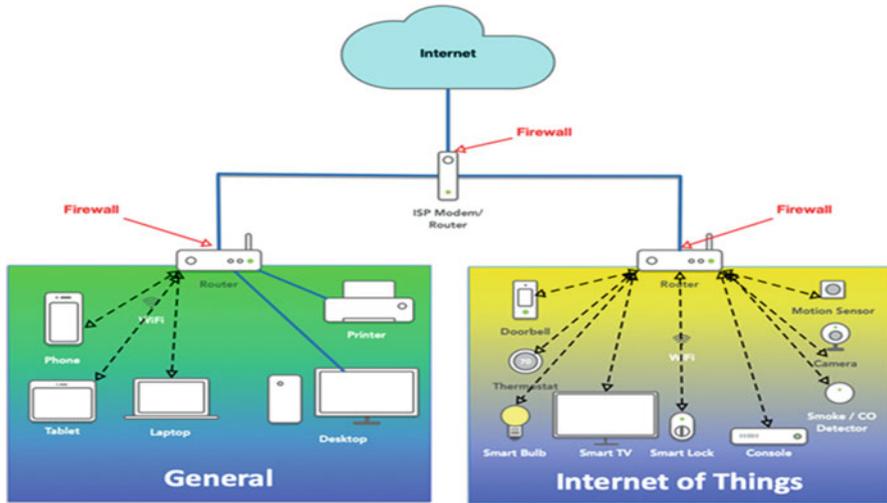


Fig. 3 Routers could be added to extend the layers of trust [11]

router on the left and the streaming is completely run through the router on the right. Network segmentation works to isolate problems [11]. If a laptop gets infected with malware, it would be able to get to the IoT network because the firewall is in front of that IoT network. It is true that if an IoT device is compromised; the firewall on the general network will defend it from malware-infected IoT devices in the same house, because of the firewall.

5.2 Secure Hardware

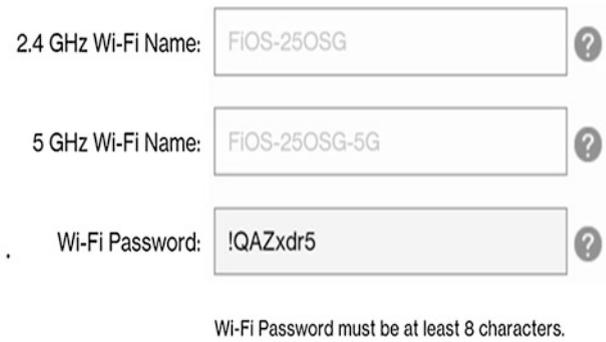
It is necessary to make the hardware secure by blocking any potential threats. It is important to update or upgrade the IoT devices when possible. It is highly recommended that one makes a long, random password, so that it is harder for hackers to guess. It is significant to use secure protocols and strong encryptions. Data should not only be encrypted when being sent, but they should also be encrypted when they are stored. Additionally, it is advised to change the default password for social models of wireless equipment. These default passwords are well known and posted on the internet and viewed by hackers.

Any attacker who comes within signal range of the unsecured router, can easily log in with the default password, if it is unchanged. Additionally, the hacker can change the password to their choice and preventing access to the owner, effectively taking over, and hijacking the network. If the user does not change the default password, a host will be in trouble, because there is a security gap between the network and everyone. Attacker can access a computer file, by using an internet

connection for illegal purposes. Attackers can introduce a different type of virus and malware to an entire network and all devices and computers connected to the home's network. Figures 4 and 5, below, demonstrate the before and after scenarios of setting a password.

Personalize your Wi-Fi settings

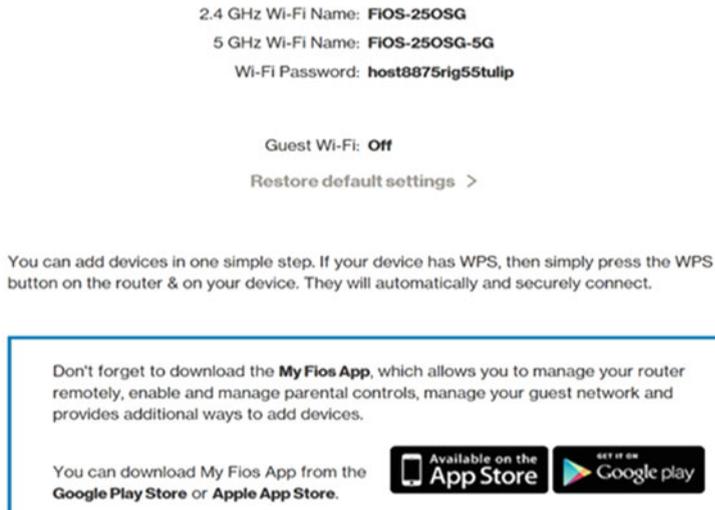
Your router is pre-configured with the Wi-Fi settings below. You may use the defaults or change the name and password to something easier to remember.



The screenshot shows three input fields for Wi-Fi settings. The first field is labeled '2.4 GHz Wi-Fi Name:' and contains the text 'FIOS-25OSG'. The second field is labeled '5 GHz Wi-Fi Name:' and contains 'FIOS-25OSG-5G'. The third field is labeled 'Wi-Fi Password:' and contains '!QAZxdr5'. Each field has a question mark icon to its right. Below the fields, a note states 'Wi-Fi Password must be at least 8 characters.'

Fig. 4 Before changing the default password

Step 3 Review and Apply your Wi-Fi settings:



The screenshot shows the updated Wi-Fi settings. The '2.4 GHz Wi-Fi Name:' is 'FIOS-25OSG', the '5 GHz Wi-Fi Name:' is 'FIOS-25OSG-5G', and the 'Wi-Fi Password:' is 'host8875rig55tulip'. Below these, 'Guest Wi-Fi:' is set to 'Off' and there is a 'Restore default settings >' link. A text block below explains that devices can be added in one step using WPS. At the bottom, there is a box with text: 'Don't forget to download the My Fios App, which allows you to manage your router remotely, enable and manage parental controls, manage your guest network and provides additional ways to add devices. You can download My Fios App from the Google Play Store or Apple App Store.' To the right of this text are logos for the App Store and Google Play.

Fig. 5 Results after changing the default password

Figure 6, below, demonstrates how the IoT can be secured by using a router which has different ports. By using this idea, each type of device will have a specific port. The diagram shows that the access port can be used to separate the two IoT devices: the laptop and Wi-Fi security camera. Using the dynamic VLAN, it is possible to partition the network to create isolated segments. The subdivision of the network into multiple VLANs is completed by configuring the network equipment such as a router or switch. Each specific device is grouped by their respective VLANs, minimizing security threats.

5.3 VPN

A VPN offers a private connection to the network and can deny access from outside users. Most organizations select a fixed VPN. The fixed VPN is one that is normally given by a system supplier or a web access supplier (ISP). Individuals may utilize these to interface a branch office with a primary office, with the goal that the branch office would be a component of a corporate system [12]. Fixed VPNs can be helpful; thus, most organizations have them. Figure 7 indicates the “Pulse Secure” company that offers VPN. The “Pulse Secure” creates a secure connection to the corporate. Also, it has a “Connect Secure SSL VPN” gateway to provide instant access to business [12]. An encrypted VPN service also provides a layer of encryption at the router level that helps in protecting all network traffic that enters or leaves a



Fig. 6 This shows how to secure all IoT appliances

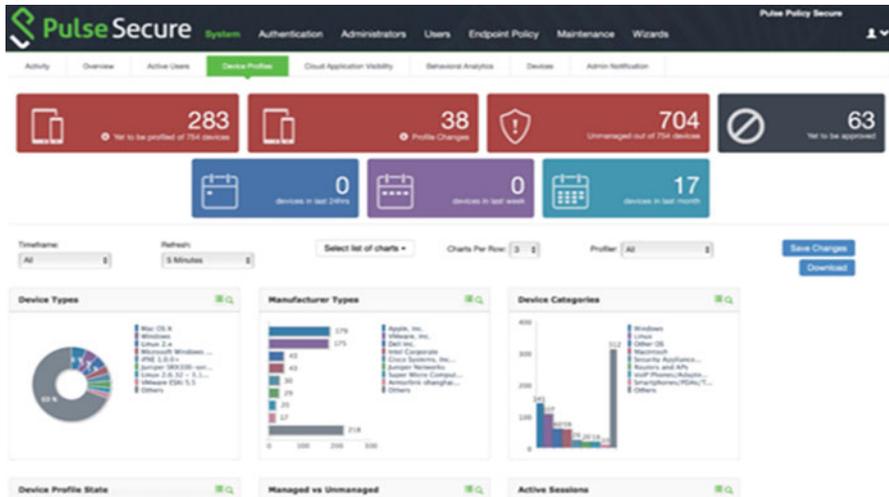


Fig. 7 Pulse secure VPN [12]

computer or device. This makes it difficult for hackers to break into your computer or device and steal information.

5.4 Virtual Local Area Networks (VLANs)

A large business enterprise likely involves elaborate network segmentation techniques because of IoT edge computing and other processing done outside of the data center [13]. However, small businesses must focus on the creation of a virtual local area network (VLAN) which should suffice and improve network security dramatically. Rather than requiring a new switch for each LAN, VLANs allow one switch to logically segment multiple networks. This allows an administrator to set different permissions and security parameters for devices hosted on each VLAN. Figure 8, below, shows how to design the VLANs. Though basic VLANs are relatively easy to design using tutorials, which are found throughout the web, things get more complicated when using tags to route traffic across VLANs. Also, if the business offers guest Wi-Fi, it is necessary to create a separate VLAN specifically for that public traffic. To ensure this network, the hardware must support VLANs.

The objective of device isolation is to form isolated network segments inside the internal private network and among the external Internet worlds. Several techniques are used to facilitate network segmentation [11]. Moreover, there are two different approaches to achieve device isolation. Segmentation is a lucrative and scalable alternative to a static VLAN method, because it allows for defense of the IoT systems within and between cells at layers one and two of the Purdue model [11].

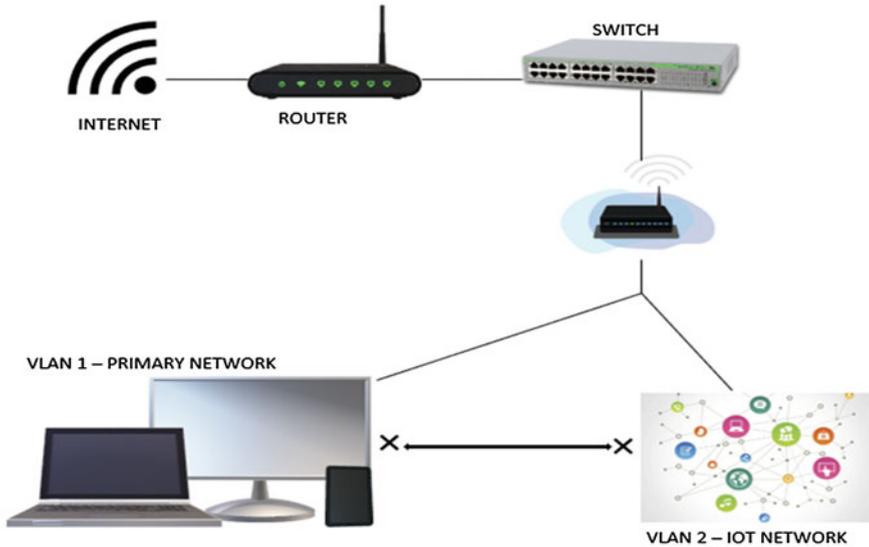


Fig. 8 Network segmentation [11]

This is because the defense mechanism in the Purdue model is more effective with segmentation. This technology is a model for implementing security at the switch level between IoT devices on the same switch between cells and creates the desired segmentation through the Industrial zone [11]. The isolated VLAN “IoT Network” to which altogether physical devices are connected. VLAN delivers the capability to limit network services to users who are members of the VLAN, essentially safeguarding applications and services. Most network equipment supports up to 4096 VLANs, which should be more than enough for most all IoTs; nevertheless, in reality, it can be a challenge and necessity to be architected properly [11].

6 Conclusion

In conclusion, the use of IoT is expanding in different businesses, which provides the benefits to companies. Generally, the Internet of Things (IoT) is regarded as the next evolution of the internet that is attracting companies. There are several benefits of utilizing IoT, such as data sharing, tracking employees, and customer engagement in business. This puts new risks on the business and new difficult obstacles to face. It is necessary to understand the challenges like privacy, limitations, security, and all the things that IoT contain. Therefore, businesses are asked to follow the recommendations presented in this chapter, which will mitigate the risks that may occur in the future. These recommendations include assessing risk, monitoring,

adding security, segmenting networks, securing hardware, and using VPNs and VLANs.

References

1. C. Liu, C. Yang, X. Zhang, J. Chen, External integrity verification for outsourced big data in cloud and IoT: A big picture. *Future. General. Comput. Syst.* **49**, 58–67 (2015)
2. W. Aman, E. Snekenes, Managing security trade-offs in the internet of things using adaptive security, in *In 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, (2015), pp. 362–368
3. S. Alharby, N. Harris, A. Weddell, J. Reeve, The security trade-offs in resource constrained nodes for IoT application. *Int. J. Electr. Comp. Eng. Electr. Commun. Eng.* **12**(1), 52–59 (2018)
4. H. Guo, J. Ren, D. Zhang, Y. Zhang, J. Hu, A scalable and manageable IoT architecture based on transparent computing. (2017), Retrieved from <https://www.sciencedirect.com/science/article/abs/pii/S0743731517302150>
5. Z. Ji, Q. Anwen, The application of internet of things (IOT) in Emergency management system in China, in *Technologies for Homeland Security (HST), 2010 IEEE International Conference on*, (IEEE, 2010), pp. 139–142
6. S. Suha Ibrahim, Security Challenges And Limitations In Iot Environments. [online] [Paper.ijcsns.org](http://paper.ijcsns.org). (2019) Available at: http://paper.ijcsns.org/07_book/201902/20190224.pdf. Accessed 11 Mar 2020
7. N.M. Kumara, P.K. Mallick, Blockchain technology for security issues and challenges in IoT. *Procedia Comp. Sci.* **132**, 1815–1823 (2018)
8. I. Lee, K. Lee, The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Bus. Horiz.* **58**(4), 431–440 (2015)
9. O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, P. Doody, Internet of things strategic research roadmap. *Int. Things-Global Technol. Soc. Trends* **1**(2011), 9–52 (2011)
10. D. Balaban, How to Set up a Secure Home Network. Medium. (2019). <https://medium.com/hackernoon/how-to-set-up-a-secure-home-network-a3d0f829fd6c>
11. C. Davis, Home Network: A Must in the IoT Era. (2018), Retrieved from <https://www.ckd3.com/blog/2018/10/15/home-network-segmentation-a-must-in-the-iot-era>
12. Orchestration. (n.d.). Retrieved 23 Mar 2020, from <https://www.pulsesecure.net/>
13. H.F. Atlam, A. Alenezi, A. Alharthi, R. Walters, G. Wills, Integration of cloud computing with internet of things: challenges and open issues, in *2017 IEEE International Conference on Internet of Things, IEEE Green Computing and Communications, IEEE Cyber, Physical and Social Computing and IEEE Smart Data*, (2017), pp. 670–675

Trusted Reviews: Applying Blockchain Technology to Achieve Trusted Reviewing System



Areej Alhogail, Ghadah Alhudhayf, Jood Alanzy, Jude Altalhi, Shahad Alghunaim, and Shahad Alnasser

1 Introduction

In many business applications, the members' opinion is crucial and can play a significant role in business development and success. Reputation systems give the authority to members to write their own reviews on online business applications to encourage other members to trust the business [1]. Reviews help a business discover people's opinion and experiences about their products or services. Moreover, they have a significant impact on people's choices and selections. Nevertheless, the current processes of business services' reviewing system raise concerns around fairness, and accuracy. Products' or services' reviews may suffer from inaccuracy due to several factors, such as competitors who may be motivated to write fake negative reviews to affect the company's reputation and lower revenues. On the other hand, members may also write positive fake reviews [2], which makes it challenging to find trusted, authentic reviews and experiences. Therefore, to improve the quality and reliability of reviews, Blockchain technology can be applied to improve the quality of submitted reviews. Blockchain is a new emerging technology that has received remarkable attention in many fields including Reputation systems [3]. In simple terms, Blockchain can be defined as a chain of blocks that contains timed stamp information, so that it is not possible to backdate them or tamper with them. It runs on a peer-to-peer system and serves as an immutable ledger that allows transactions to take place in a decentralized manner [4]. Blocks are recorded on a public ledger that can be seen by anyone on the chain [3]. Moreover, reputation is not portable. For instance, marketplaces such as Amazon and eBay have developed their review system to enable suppliers to build their reputation over time. The system,

A. Alhogail (✉) · G. Alhudhayf · J. Alanzy · J. Altalhi · S. Alghunaim · S. Alnasser
King Saud University, Riyadh, Saudi Arabia
e-mail: Aalhogail@ksu.edu.sa

unfortunately, is locked and closed, which means that submitting reviews on eBay is meaningless on Amazon [2]. In addition, today's business applications require a trusted platform to exchange member reviews regardless of the business product. So, Blockchain technology would facilitate such a trusted platform [5].

Many features of the Blockchain make it applicable to be used. For example, the access control can be incorporated into the Blockchain through unique identifiers, so all participants are known [6]. In addition, there is the consensus property of the blockchain that requires the majority of miners to validate a block before its inclusion into the chain [7, 8]. Once a block is added after consensus, it cannot be changed or removed even by its original author [8]. To achieve integrity, each block must be digitally signed with the owner's private key and assigned a hash [3], increasing the possibility of having authentic and trustworthy reviewers. Moreover, the immutable ledger that is tamper resistant ensures more integrity [7]. Finally, each block maintains a reference to its parent block and a timestamp which can facilitate Auditability.

This paper aims to propose a model that implements blockchain technology to improve the authenticity and quality of reviews submitted to the system and make it trust worthier. The model will be tested by applying it on a holiday home website to ensure its effectiveness. Members can read and write a review relating to any available property that matches the user's search criteria. They can also like or dislike any review to ensure the agreeableness with it, based on the like and dislike ratio. Thiqah – Trust Credit – will be given to review's writer as an incentive, serving a significant role in our reward system to encourage them to write legitimate reviews. It will enhance the quality of the reservation process and enable members to choose a suitable home in a convenient way. Thiqah is an Arabic word that means trust. Consequently, more genuine reviews will be submitted, improving the system's reputation legitimacy.

The remainder of this paper is organized as follows. First, a literature review of the investigated topic is presented. Then, it is followed by describing the Trusted Review model and methodology used to collect supportive data. The data and system requirements are then analyzed to describe the implementation of the testing system. Finally, the conclusion and future studies are presented.

2 Literature Review

2.1 Reviews and Reputation Systems

Reviews are the evaluation and description of a user's experience with a product or service, such as staying at a hotel. Reputation measures how much the community trusts a user. It is based on the user's previous transactions and interactions with the community. The greater a user's reputation is, the more trustworthy that user is

perceived on the network [9]. Reputation has a great influence on many businesses, for instance, Airbnb and Uber are building trust through ratings and reviews [8].

Reputation system collects, maintains, and disseminates reputations—aggregated records from past interactions—of each participant in a community. Reputation systems have been designed for use in many settings, including online auctions, e-storefronts, and a wide range of peer-to-peer systems [10]. They provide one of the most successful incentive mechanisms, and reputation systems are widespread on the internet today. Negative reviews can seriously affect the reputation of a business. The lack of a transparent communication mechanism between websites offering review systems and end-users also creates a loophole that allows products’ or service providers to easily deal with the current rating system manager and thereby change review scores in their favor [10].

Reputation has many characteristics. First, there are context-specific reputation systems; for example, Mike trusts John as his doctor, but he does not trust John as a mechanic who can fix his car. So, in the context of seeing a doctor, John is trustworthy. But in the context of fixing a car, John is untrustworthy. Second, there are multifaceted reputation systems. A member might evaluate a restaurant from several aspects; for example, the quality of food, the price, or the service. For each aspect, they develop a kind of trust. The overall level of trust depends on the combination of the member’s trust for each aspect. Finally, there are dynamic reputation systems, in which a reputation might increase or decrease with the further experiences of other reviewers [10].

There are four threats to the integrity of reputation systems:

- **Whitewashing:** In this practice, a person creates a new name and starts over with a clean reputation.
- **Incorrectly reported feedback:** This is caused by conflicts of interest or a reviewer’s desire to improve others’ perception of them may lead reviewers to report incorrect feedback.
- **Phantom feedback:** This refers to the submission of a review that never happened, such as reviewing a hotel a reviewer never stayed at.
- **Sybil attacks:** In these attacks, a single agent creates many fake online identities to boost the reputation of their primary online identity [11].

Most reputation systems have been implemented using the centralized server model on multiple web services that makes control done by a single entity affecting issues related to trust and reputation [9].

2.2 Applying Blockchain in Reputation System

Blockchain is a distributed database solution that maintains a continuously growing list of data records that are confirmed by a subset of network participants (also known as “miners”) who enrich the chain by solving difficult computational problems [12]. Information about every transaction completed in the blockchain

is shared and available to all nodes, which does not require any third-party organization through the elimination of transactional costs [13].

Autonomy, self-sufficiency, and decentralization are considered to be important elements to differentiate trusted reviews. Autonomy occurs after the contract is launched and running, and no further contact will be available between the contract and its initiating agent. Self-sufficiency can be described as the ability to collect resources such as raising funds by providing services or issuing equity, and spending them on needed resources. Since smart contracts play an important role in the Blockchain they share the decentralization characteristic, they are distributed across multiple nodes [14].

The advantage of blockchain is that the public ledger cannot be modified or deleted after the data have been approved by all nodes (i.e., consensus), meaning that once a transaction has been added to a block, which is in turn added to the blockchain, this transaction cannot be altered or deleted [12]. This is why blockchain is well known for its data integrity and security and why it facilitates the exchange of information in a way that provides all involved parties with access to a transparent and shared database [15]. This feature makes it attractive for the proposed application, which is aimed toward achieving reliable and trusted reviews about any applicable product or service.

The goal of a reputation system is to make it possible for people to share their opinions and experiences on goods and services. However, there are several fundamental problems that remain unsolved. One of these problems is how participants can be sure that the reviews are not being edited or deleted [11]. Applying blockchain technology in a reputation system ensures feedback legitimacy. Fragments of reviews are saved in the blockchain, which guarantees that reviews cannot be edited or deleted later.

There are some applications available in the literature that implement blockchain technology. Firstly, Revain is the first review platform to implement blockchain technology in ensuring that all reviews are genuine and legitimate. Fragments of reviews are saved in the blockchain, which guarantees that those comments cannot be edited or deleted [16]. The platform makes use of artificial intelligence, which filters out low-quality reviews and makes high-quality ones eligible for rewards. It uses Ravencoin (RVN) which is a stable token that is not subject to volatility and exists only inside the Revain platform. Revain is also transparent: all reviews are saved permanently and cannot be changed or removed by anyone. Any user has the opportunity to view the list of transactions in the blockchain and edit them [16].

LINA.Review is a blockchain-based review platform that utilizes blockchain's immutability for the greatest possible transparency, to create conditions for reviewers to benefit from providing quality reviews, and to easily and directly interact with users and providers of products or services that are currently trustless [17]. LINA.Review introduced a set of criteria applied to different fields, including technological products, hotels, movies, and books. With this set of criteria available, users can submit more accurate reviews through the LINA rating app (available on Google Play) [17]. The above-mentioned system falls short of providing Arabic support which is an initial requirement for many Arabic users.

3 Methodology

3.1 Trusted Reviews Model

In order to achieve reliable reviews, a model of Trusted Reviews is suggested. The main goal of the proposed model is to deliver authentic and immutable members' reviews and feedback by applying the Blockchain technology, where members can write and/or read reviews. Moreover, it should implement a reward system to motivate website members to submit reviews, as shown in Fig. 1.

At first, the member writes a review about a property, the review is then passed to the smart contract so that it can be added to the blockchain. A smart contract can be defined as the Blockchain transactions which can involve more embedded detailed instructions, eliminating the need of a third party [14]. The use of a smart contract enables the member to interact with the Blockchain interchangeably such as retrieving reviews. A review can be rated by calculating the ratio between likes and dislikes. If a certain ratio is achieved, the writer of the review will earn Thiqah credit in return for his/her trustworthy review.

For the review blockchain, each block consists of the block header and the block body. Each block contains a block ID and a block hash to identify the block. To add Auditability, each block maintains a reference to its previous block and a timestamp.

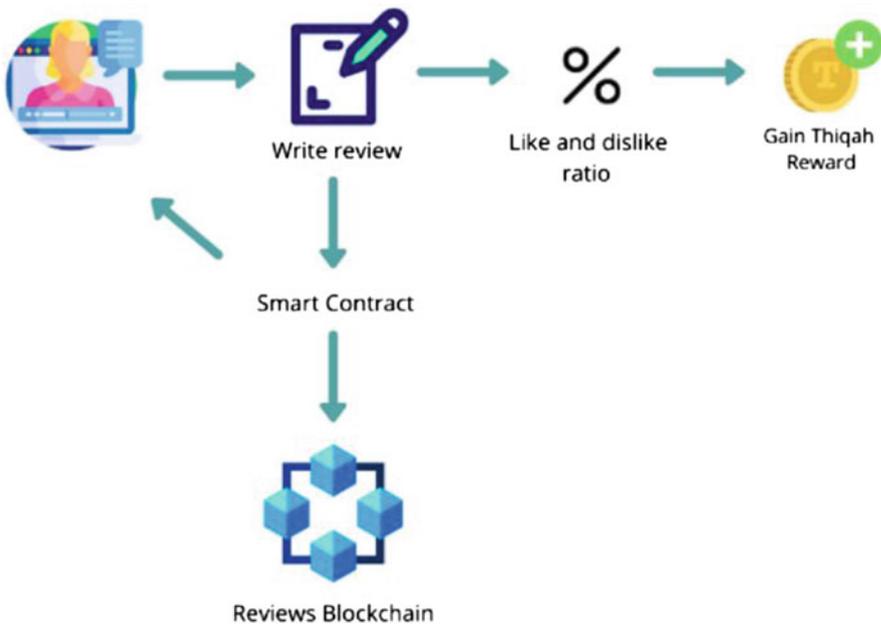


Fig. 1 Trusted reviews model

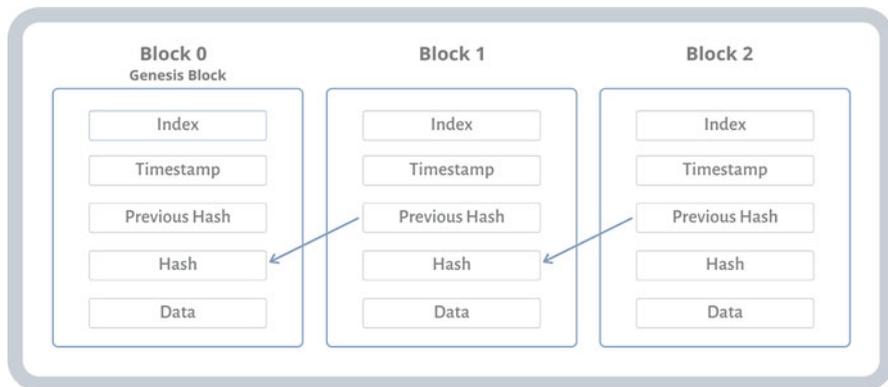


Fig. 2 Blockchain architecture

Once a block is added after consensus, it cannot be changed or removed even by its original author (Fig. 2).

As the blockchain technology is based on a peer-to-peer network, when the transaction is created by the user, they broadcast it to all peers in the network [12]. Miners collect broadcasted transactions and attempt to merge them into a block that satisfies a cryptographic hash function. The technology also secures interactions between two individuals by using public-key cryptography, whereby each agent is assigned a private key that is kept secret like a password and a public key that is shared with all other agents [18].

There are three types of blockchain [3]: *Public* blockchain, where all records are visible to the public and every person, could participate in the consensus process. In *Private* blockchain, only specific nodes can participate in the consensus process. The *consortium* blockchain has a decentralized authority and comes with mixed properties from public and private blockchain. We have used a consortium with public reading and private writing.

A block is valid if it has valid transactions, as soon as miners receive the broadcast, they take several transactions to confirm that they are legitimate and put them into a block. The miner who adds the next block to the blockchain is the first to have assembled a valid block and has found a valid solution to the hash function. This specific mining technique is called proof-of-work; it is the original consensus algorithm implemented in a blockchain network. This algorithm is used to confirm transactions and add new blocks to the chain [18–20].

Although the aim is to suggest a general reputation system that can be implemented into any e-commerce application, we will focus on the application of this system on a holiday home website that has been developed to test the proposed model. By using smart contracts in Trusted Reviews, we can ensure that no property exists twice. Also, each review is linked to a property through the smart contract to add authenticity to the reviews and increase trustworthiness. With the use of contracts, common problems can be solved by taking out the human judgment which

usually makes things run smoothly [14]. Since we are using public reading and private writing, anyone can read the reviews, but only registered members can write.

3.2 Data Collection

In order to clarify the interest and requirements for a trustworthy reviewing system, this project applies a quantitative method of user opinions' survey. The survey was designed to gather information about the importance of the proposed project and how it could improve users' decision-making in booking holiday homes. Holiday homes have been selected to narrow the field of study and make the reflective responses. The survey was created using Google forms and distributed through social media platforms such as WhatsApp and Twitter. It collects data about previous user experiences and their interest in applying blockchain in a reviewing system. It consists of 12 questions that vary between 3- and 5-point Likert-type.

4 System Analysis Results

4.1 Quantitative Data Analysis

Almost 300 responses were collected and analyzed. It had been determined that 84.6% of members are using online booking services. The majority had had an unreliable experience after trusting these reviews, which motivated us to apply this technology to provide a solution to this problem. Moreover, approximately 72% of the applicants had never heard of blockchain technology before. On the other hand, 90% were interested in using a new technology to ensure trustworthy reviews. Most of the respondents believe that Arabic support is essential to them to effectively use the system.

Results show that a reviewing system that implements Blockchain technology to achieve trusted, authentic, and real reviews is needed. These review characteristics will be determined based on the previous transactions/history of the member that are stored in the blockchain in an immutable way. A reward system will be used to mitigate trust issues in finding suitable holiday homes based on members' preferences.

4.2 System Requirements

In order to test the model, a holiday home website has been designed in accordance with the requirements collected from the survey. Trusted Reviews website is

designed to provide a trustful platform targeting property owners to offer their holiday homes and members' contributions in sharing experiences. It supports English language. Moreover, it does not require professional technical skills.

There are two types of users: holiday homeowner, who is the service provider. The other is the member, who is the regular user of the system. A list of each user system requirements is as follows:

- Property owner
 - System shall allow the property owner to add/remove/ modify property
 - System shall allow the property owner to read reviews.
- Member
 - System shall provide the user the ability to search with suggestions based on search criteria
 - System shall allow the member to read/write reviews.
 - System shall allow the member to like or dislike other member's reviews.
- System
 - System shall provide Thiqah to members with useful reviews.
 - System shall view top five facilities/service providers based on the highest scores.
 - System calculates the overall score of service provider based on the sum of stars divided by the number of reviews.
 - System shall view top three reviews based on Thiqah incentives.

4.3 System Design

The system use case diagram is displayed in Fig. 3.

Trusted Reviews architecture, as shown in Fig. 4, provides higher security through the layered system, which increases flexibility and maintainability through the clear, well-defined structure, which consists of six layers: User Interface Layer that includes three interfaces: Admin, Property Owner, and Member; Application Layer that consists of main functionality; Authorization Layer that is responsible for authorizing users in order to assign suitable privileges; Blockchain Layer that is in control of storing the most important data into the block where it has to be immutable; Data Access Layer that provides easier access to data stored in the database; and Physical Layer where it connects with different databases [21].

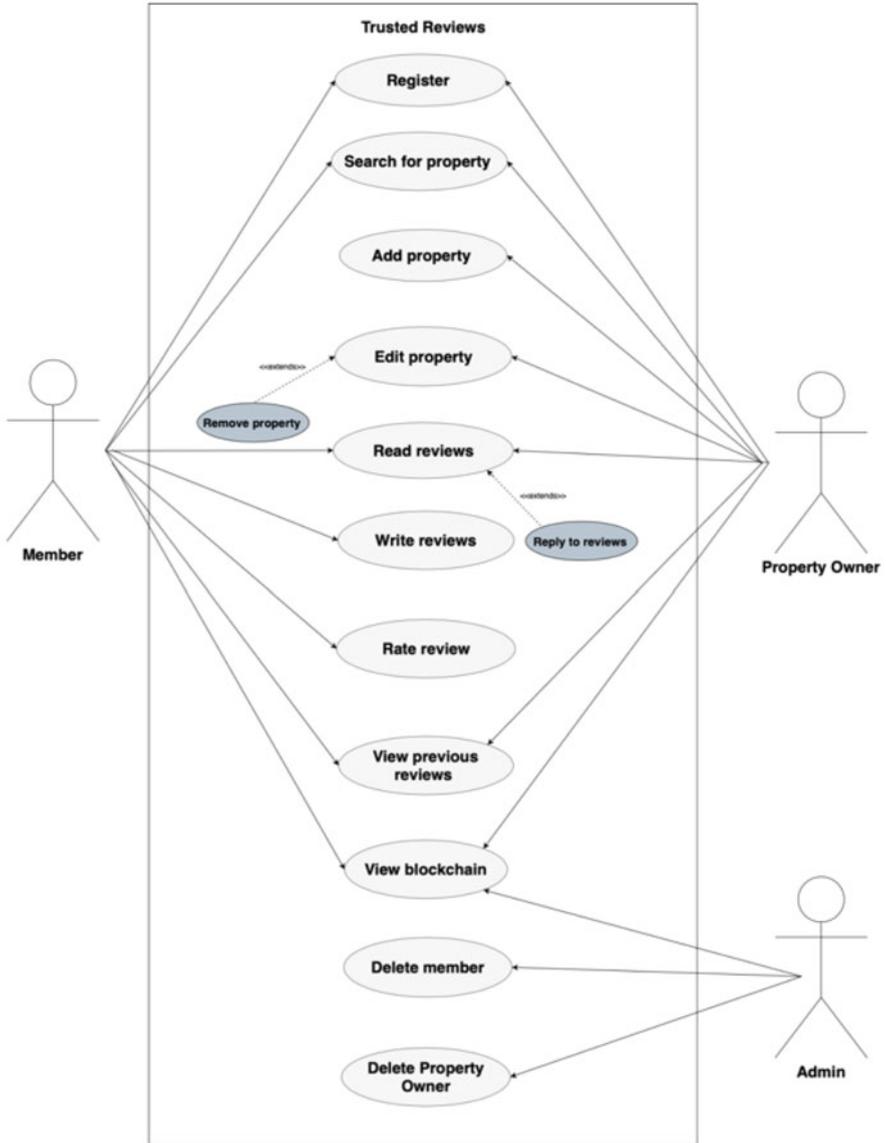
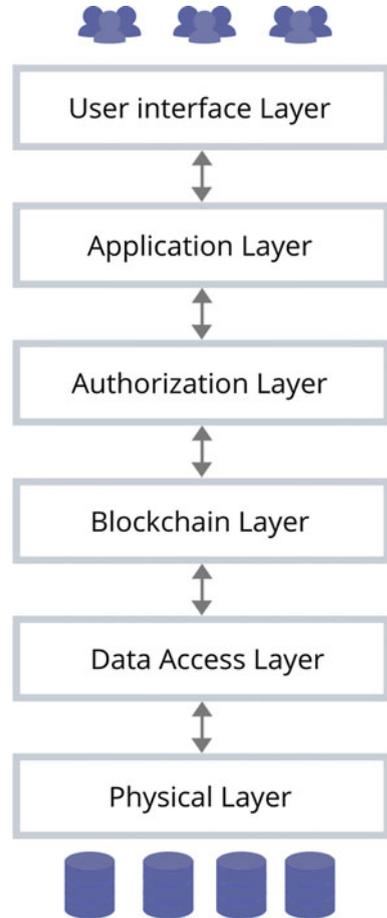


Fig. 3 Use case diagram

Fig. 4 Trusted reviews architecture



5 Implementation

In order to deploy blockchain technology to Trusted Reviews, it started by creating the smart contract which was written in solidity programming language that is provided by Ethereum public blockchain. The smart contract plays a role in processing the user’s transactions to create a block.

Ganache truffle suite provides developers with fake Ethers to be able to develop and interact with the blockchain. It shows all constructed blocks and their detailed information such as, block hash, timestamp, and gas used in each transaction.

Following the block creation, a transaction confirmation is requested through MetaMask. Downloaded as a Google chrome extension, it acts as a manager of the virtual wallet. It stores all transactions whether confirmed or rejected and transaction

gas fees. A block is then added to the blockchain after the transaction had been confirmed by the user.

The smart contract included Trusted Reviews' main functionalities such as adding Property and add Review. A sample code is provided for each function.

5.1 Code Snippets

```
function addingproperty(uint propertyId, string memory
propertyname, string memory propertytype, string memory
propertydesc,
    string memory propertyaddress, string memory propertycity)
    public {
        require(properties[propertyId].isExist == false,
            "Property already exists");
        properties[propertyId].Id = propertyId;
        properties[propertyId].propertyowner = msg.sender;
        propertiesdet[propertyId].propertyname = propertyname;
        propertiesdet[propertyId].propertyaddress =
            propertyaddress;
        propertiesdet[propertyId].propertytype = propertytype;
        propertiesdet[propertyId].propertydesc = propertydesc;
        propertiesdet[propertyId].propertycity= propertycity;
        properties[propertyId].isExist = true;
        propertyCount+=1;
        propertyIds.push(propertyId);
    }function addreview(uint propertyId, string memory review)
public{
    properties[propertyId].reviews.push(review);
    properties[propertyId].reviewCount++;
}
}
```

Mapping to store property and property details in arrays, to be able to link reviews to them, and to keep track of total number of properties.

```
//storing property
mapping(uint=> Property) public properties;
//storing property details
mapping(uint =>PropertyDetails) public propertiesdet;
//properties count
uint public propertyCount;
uint [] propertyIds;
```

To connect the blockchain with the front-end, Web3j has been used in addition to adding some script tags on the Hypertext Markup Language (HTML) documents as follows:

```
initWeb3: async function() {
    // Modern dapp browsers...
    if (window.ethereum) {
        App.web3Provider = window.ethereum;
```

```

try {
  // Request account access
  await window.ethereum.enable();
} catch (error) {
  // User denied account access...
  console.error("User denied account access")
}
}
else if (window.web3) {
  App.web3Provider = window.web3.currentProvider;
}
// If no injected web3 instance is detected, fall back
to Ganache
else {
  App.web3Provider = new Web3.providers.HttpProvider
  ('http://localhost:7545');
}
web3 = new Web3(App.web3Provider);
return App.initContract();
}, <script src="js/web3.min.js"></script>
<script src="js/truffle-contract.js"></script>
<script src="js/app.js"></script>

```

5.2 Conclusion and Future Studies

Achieving trustworthy reviews is a demand for today's electronic-based business environment. People tend to base their selections on others' experiences and reviews. Therefore, everyone is looking for a trusted platform to exchange feedback. This paper has proposed a Trusted Review model that utilizes blockchain technology to enhance the trustworthiness of reviews by applying smart contract to monitor the rules and taking out the human judgment factor. Moreover, the reward system has been used to motivate members to write trusted and honest reviews, which is a ratio of likes and dislikes, used to increase trust factor of each member through a credit point system named Thiqah (the Arabic word of Trust). Reviews are only added after the consensus of other members.

The model has been applied on a holiday home website to investigate the application and effectiveness of the model. The testing of the website showed that the use of blockchain is worthy and effective in testing environment. More investigation will be taken to study the use of the model with other e-business applications and with more users. Moreover, we are planning to investigate more accurate algorithms for calculating reputation scores. There are some limitations in the deployment and use of blockchain in reputation systems. Some are due to basic flaws in the blockchain protocol architecture. For example, time required to mining and a maximum block size.

References

1. G. Koynov, Could a Blockchain Based Reputation System Prevent a Dystopian Future?, 2019. [Online]. Available: <https://hackernoon.com/could-a-blockchain-based-reputation-system-prevent-a-dystopian-future-d58522b88e2c>. [Accessed: 15-Sep-2019]
2. T. Carden, Why People Give Fake Online Reviews | 100Reviews, 2019. [Online]. Available: <https://www.100reviews.com/why-people-give-fake-online-reviews>. [Accessed: 15-Sep-2019]
3. Z. Zheng, S. Xie, H. Dai, X. Chen, H. Wang, An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. Proc. - 2017 IEEE 6th Int. Congr. Big Data, BigData Congr. 2017, 557–564 (2017)
4. S. Huckle, R. Bhattacharya, M. White, N. Beloff, Internet of things, blockchain and shared economy applications. Procedia Comput. Sci. **58**, 461–466 (2016). <https://doi.org/10.1016/j.procs.2016.09.074>.
5. S. Lee, A Decentralized Reputation System: How Blockchain Can Restore Trust In Online Markets, 2018. [Online]. Available: <https://www.forbes.com/sites/shermanlee/2018/08/13/a-decentralized-reputation-system-how-blockchain-can-restore-trust-in-online-markets/#7e48b0e8481a>. [Accessed: 17-Dec-2019].
6. M. Swan, *Blockchain: Blueprint for a New Economy* (O'ReillyMedia, Inc., 2015)
7. H. Al-megren, S. Alsalamah, S. Altoaimy, L. Alsalamah, L. Soltanisehat, Blockchain Use Cases in Digital Sectors: A Review of the Literature, in *2018 IEEE Confs on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, Congress on Cybermatics*, (IEEE, 2018), pp. 1417–1424
8. M. Sharples, J. Domingue, The blockchain and kudos: A distributed system for educational record, reputation and reward mike. Adapt. Adapt. Learn. **2016**, 490 (2016)
9. R. Dennis, G. Owen, Rep on the block: A next generation reputation system based on the blockchain. 2015 10th Int. Conf. Internet Technol. Secur. Trans. ICITST 2015, 131–138 (2016)
10. Y. Wang, J. Vassileva, Trust and reputation model in peer-to-peer networks. Proc. - 3rd Int. Conf. Peer-to-Peer Comput. P2P 2003 **October 2003**, 150–157 (2003). <https://doi.org/10.1109/PTP.2003.1231515>
11. E. Friedman, P. Resnick, R. Sami, Manipulation-Resistant Reputation Systems
12. J. Yli-Huumo, D. Ko, S. Choi, S. Park, K. Smolander, Where is current research on Blockchain technology? – A systematic review. PLoS One **11**(10), 1–27 (2016). <https://doi.org/10.1371/journal.pone.0163477>.
13. M. Pilkington, Blockchain technology: Principles and applications. Res. Handbooks Digit. Transform., 225–253 (2016). <https://doi.org/10.4337/9781784717766.00019>
14. C. Smith, Blueprints for a new economy. **293**(11) (2011)
15. S. Seebacher, R. Schüritz, Blockchain technology as an enabler of service systems: A structured literature review BT – Exploring services science. Explor. Serv. Sci. **279**(Chapter 2), 12–23 (2017). <https://doi.org/10.1007/978-3-319-56925-3>
16. Revain, 15 – Revain White Paper.pdf.
17. L. Network, LINA.NETWORK (LINA) Blockchain Based Application for Innovation, 2019
18. R. Dennis, G. Owenson, “Rep on the Roll: A Peer to Peer Reputation System Based on a Rolling Blockchain, (2016)
19. S. Dexter, How Are Blockchain Transactions Validated? Consensus VS Validation, 2018. [Online]. Available: <https://www.mangoresearch.co/blockchain-consensus-vs-validation/>. [Accessed: 30-Sep-2019]
20. A. Tar, Proof-of-Work, Explained, 2018. [Online]. Available: <https://cointelegraph.com/explained/proof-of-work-explained>. [Accessed: 30-Sep-2019]
21. R. Elmasri, S. B. Navathe, The Relational Data Model and Relational Database Constraints, no. September 2019. 2016

Large-Scale Parallelization of Lattice QCD on Sunway TaihuLight Supercomputer



Ailin Xu, Zhongzhi Luan, Ming Gong, and Xiangyu Jiang

1 Introduction

1.1 Lattice Quantum Chromodynamics

Quantum field theory is a theory describing the composition and interaction of known material worlds. Quantum chromodynamics (QCD) is a theory describing strong interactions in quantum field theory. Currently, lattice quantum chromodynamics (lattice QCD (LQCD)) is the only effective method for quantum chromodynamics research. Therefore, lattice QCD has strong research significance.

The lattice QCD theory proposed by Wilson in 1974 [1] has developed into a mainstream non-perturbative theoretical calculation method of QCD. QCD is a kind of field theory. Field theory allows infinite degrees of freedom, so coordinate values are continuous. But in lattice QCD, the degree of freedom is limited, and a four-dimensional Euclidean lattice is defined to replace continuous space-time. Quark is defined at the intersection of the lattice (called grid point), and gluon is defined between adjacent grid points. When the distance between the grid points is close to 0, since QCD has the characteristics of asymptotic freedom, expected results can be obtained from the comparison between the calculation results and the experiment results.

When Wilson introduced the lattice QCD theory [1] in 1974, due to its large amount of high-precision calculation, computers at that time could not meet its

A. Xu · Z. Luan (✉)

School of Computer Science and Engineering, Beihang University, Beijing, China
e-mail: xuailin@buaa.edu.cn; 07680@buaa.edu.cn

M. Gong · X. Jiang

Institute of High Energy Physics, Chinese Academy of Sciences, Beijing, China
e-mail: gongming@ihep.ac.cn; jiangxiangyu@ihep.ac.cn

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_12

133

requirements, which seriously blocked the development of related disciplines. In recent years, new computer architectures continue to appear, providing a variety of software and hardware environments that support large-scale parallel computing. Today, the calculation in lattice QCD has become a reality with the help of modern supercomputers.

At present, scientists have achieved fruitful results on lattice QCD researching. In addition to the research of high-energy physics theory, lattice QCD has good parallelism, and it is often used by computer scientists for large-scale parallel computing tests. Scientists have tried multiple parallelization methods on different computing platforms, conducted a lot of tests, and achieved good results. Pavlos Vranas and others have implemented parallel accelerated computation of lattice QCD on BlueGene/L supercomputer and won the 2006 Gordon Bell Prize [2]. Gupta and others have implemented a lattice QCD calculation by using GPU and OpenACC framework in [3] and achieved a 5.6 times performance improvement over the CPU version. Kanamori and others compared and summarized multiple attempts on single instruction multiple data (SIMD) machines in [4], which showed good parallel results on Oakforest-PACS and Skylake-SP. In [5], a parallelization method of lattice QCD single core group on the Sunway TaihuLight supercomputer is implemented. Compared with the original serial method, the performance is improved by 63 times. It deserves further research to achieve high-performance parallel computing on multiple core groups.

1.2 SW26010 Heterogeneous Many-Core Processor

In recent years, the development of general computer architecture has been limited. The storage wall, power wall, and frequency wall have limited the computing performance of general computers [6], and cannot meet the high-precision computing requirements of lattice QCD. New heterogeneous computer architecture has better parallel computing capabilities and provides a new computing environment for a large number of high-precision scientific calculations. High-performance parallel algorithms are one of the current research hotspots.

Sunway TaihuLight supercomputer is equipped with 40960 SW26010 heterogeneous many-core processors [7], and it can provide powerful floating-point computing capabilities. Figure 1 shows the architecture of SW26010 heterogeneous many-core processor. SW26010 processor contains four core groups (260 cores in total) connected by an on-chip interconnect network, and the theoretical peak performance of double-precision floating-point calculation can reach 3.168TFlops. MPE is the management core of each core group, which is usually responsible for the communication between the core groups and the management of CPEs in large-scale parallel computing, and it does not generally handle a large number of computing tasks. The computing performance of SW26010 processor is mainly provided by CPEs on the chip. CPE's storage level is shown in Fig. 2. Each CPE has an independent 64KB local data memory (LDM) for programmers to allocate

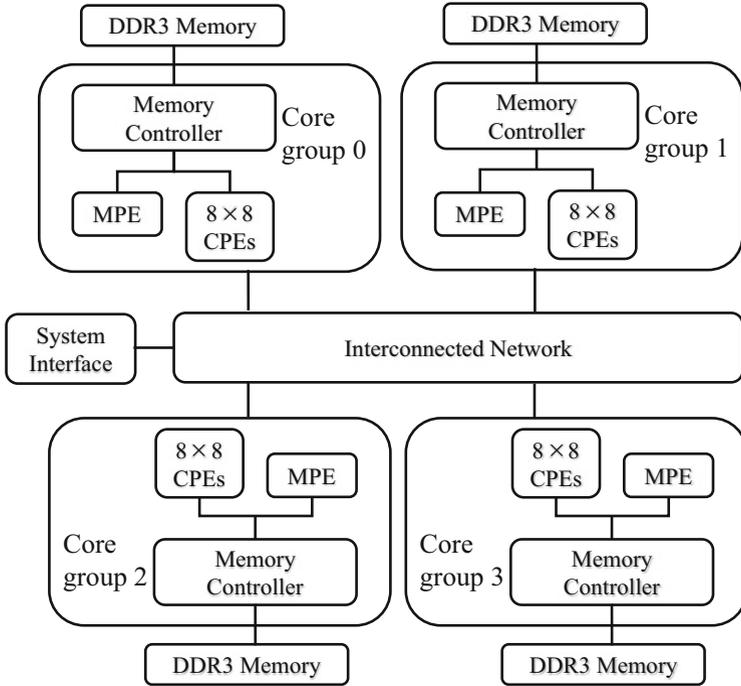


Fig. 1 The architecture of SW26010 heterogeneous many-core processor

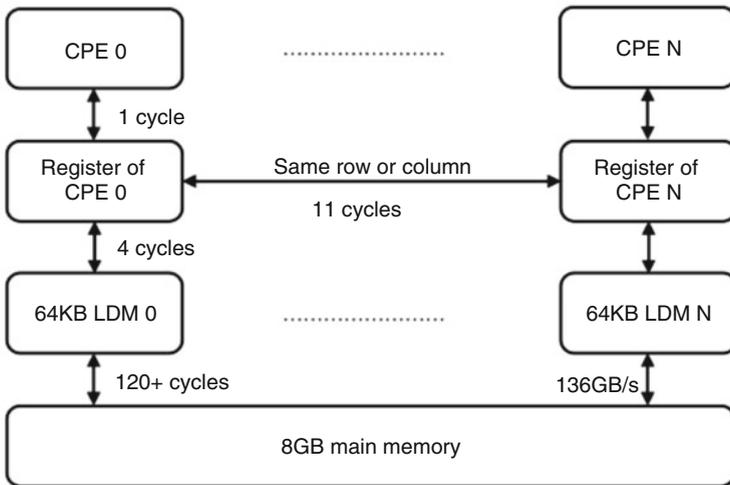


Fig. 2 The memory hierarchy of the CPEs

and can access the main memory through direct memory access (DMA). The 64 CPEs in one core group are arranged in the form of an 8×8 square matrix. CPEs in the same row or the same column support register communication, and the register communication delay is much smaller than the communication delay between the CPEs and main memory. Due to these characteristics of the SW26010 processor, optimizations aimed at LDM and data communications can often improve application performance significantly.

Many applications have been large-scaled parallelized and accelerated on Sunway TaihuLight supercomputer, including the Global Atmospheric Dynamics Simulation [8] and Earthquake Simulation [9] which won the Gordon Bell Prize in 2016 and 2017, respectively. Due to the characteristics of SW26010 processor, DMA is one of the performance bottlenecks of Sunway architecture applications. Minimizing redundant DMA data transmission and making full use of the LDM are keys to optimize and accelerate applications on Sunway architecture.

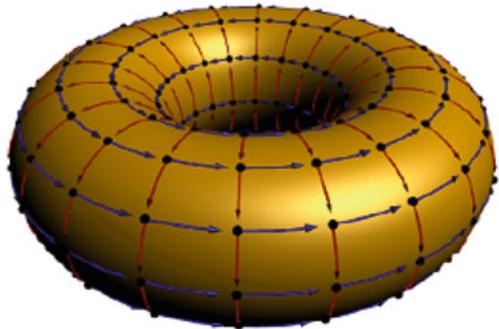
1.3 Overview of the Paper

The following structure of this article is as follows: Sect. 2 will introduce the calculation of lattice QCD. Section 3 describes the optimized data distribution method in detail and shows the parallelization method of lattice QCD application on multiple core groups. Section 4 shows our experiment and analysis of the lattice QCD application. Section 5 summarizes this paper.

2 Lattice QCD Calculation Theory

Lattice QCD theory cuts a four-dimensional space-time into four-dimensional lattice array and connects the two ends of each dimension as shown in Fig. 3. The size of this four-dimensional lattice array is not fixed, which is generally between $8 \times$

Fig. 3 Two-dimensional representation of four-dimensional lattice space



$8 \times 8 \times 8$ and $256 \times 256 \times 256 \times 512$. From the view of physical theory, the larger the grid array size is, the more accurate simulation results can be obtained. However, the amount of calculation in the Monte Carlo numerical simulation used in the algorithm doubles as the scale increases [10, 11]. That is, only high-performance parallel computing can meet the computing requirements.

The calculation theory of lattice QCD is described below. Fermion fields $\phi(x, y, z, t)$ representing quarks are placed on each lattice. It is a Grassmann number with three color components and four spin components. In the calculation, it can be represented by a vector composed of 12 complex numbers with the form (1):

$$\phi(x, y, z, t) = \begin{pmatrix} \begin{pmatrix} d_{11} \\ d_{12} \\ d_{13} \end{pmatrix} \\ \begin{pmatrix} d_{21} \\ d_{22} \\ d_{23} \end{pmatrix} \\ \begin{pmatrix} d_{31} \\ d_{32} \\ d_{33} \end{pmatrix} \\ \begin{pmatrix} d_{41} \\ d_{42} \\ d_{43} \end{pmatrix} \end{pmatrix} \tag{1}$$

Gauge fields $U_\mu(x, y, z, t)$ representing gluons are placed on the connection between adjacent grid points. It can be written as a 3×3 complex matrix with the form (2):

$$U_\mu(x, y, z, t) = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix} \tag{2}$$

Each grid point has eight such matrices connected to its eight neighbors in four dimensions. Assigning all the vectors or matrices in this four-dimensional space defines a state of this space. Through a large number of iterative calculations, the amount of Fermion fields on the grid points constantly changes and converges to a certain state. The core operation in the iteration is D-slash calculation. The D-

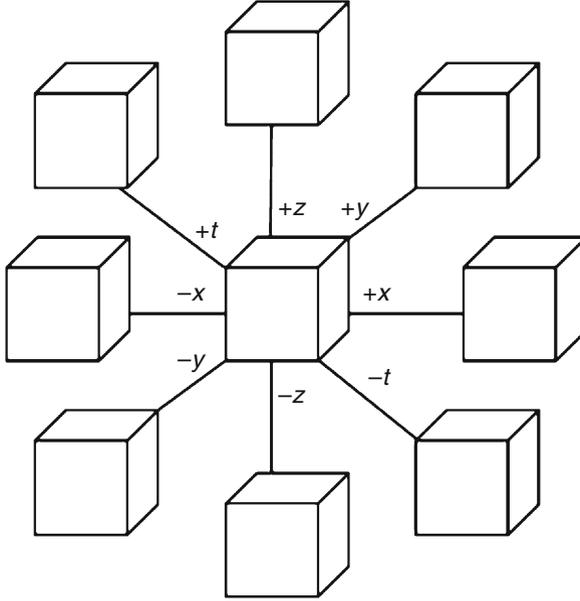


Fig. 4 Schematic diagram of four-dimensional 9-point stencil operation

slash calculation is a four-dimensional 9-point stencil operation as shown in (3) and Fig. 4:

$$\not{D} = \sum_{\mu} [(1^{(4)} - \gamma_{\mu})U_{\mu}(\mathbf{X})\delta_{\mathbf{x}+\mu,\mathbf{y}} + (1^{(4)} + \gamma_{\mu})U_{\mu}^{\dagger}(\mathbf{X} - \mu)\delta_{\mathbf{x}-\mu,\mathbf{y}}] \quad (3)$$

In (3), μ represents four space-time directions, \mathbf{X} and \mathbf{Y} are four-vector space-time coordinates, $U_{\mu}(\mathbf{X})$ represents the connection of gauge fields from point X to direction μ , U^{\dagger} is its inverted complex conjugate, and γ_{μ} is a 4×4 sparse complex matrix. One of the definitions of γ_{μ} is shown in (4):

$$\begin{aligned} \gamma_0 &= \begin{pmatrix} 0 & 0 & 0 & i \\ 0 & 0 & i & 0 \\ 0 & -i & 0 & 0 \\ -i & 0 & 0 & 0 \end{pmatrix} & \gamma_1 &= \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix} \\ \gamma_2 &= \begin{pmatrix} 0 & 0 & i & 0 \\ 0 & 0 & 0 & -i \\ -i & 0 & 0 & 0 \\ 0 & i & 0 & 0 \end{pmatrix} & \gamma_3 &= \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{aligned} \quad (4)$$

We use the minimum residual method to solve the calculation of lattice QCD. The calculation process is divided into the following steps:

- Step 1: Read the input of fermion field matrices and gauge field matrices, and then initialize the related parameters.
- Step 2: Execute D-slash calculation on each grid point, and update the value of the matrix on each grid point.
- Step 3: Calculate the global residual value.
- Step 4: If the residual value meets the accuracy requirements or the maximum number of iterations has been reached, then stop the calculation and output the updated fermion field matrix; otherwise, skip to Step 2 and continue the calculation.

Since each grid points needs to perform a D-slash operation once in each iteration and the calculation process is only related to adjacent grid points, the lattice QCD calculation process has good parallelism and large-scale scalability. The hotspot function D-slash is mainly embodied as floating-point number operations on the computer. In order to improve the program performance, optimization based on the characteristics of the target computer architecture is necessary.

3 Innovations Realized

A lattice QCD application based on a single core group of SW26010 is implemented in [5]. The paper achieved good results and also proposed some shortcomings and prospects. We optimized the application in [5] by improving the data allocation of CPEs and perform large-scale parallelization on multiple core groups.

3.1 Data Allocation of CPEs

From the further test of the application in [5], we found out that with the expansion of the parallel scale, the time proportion of data transmission, especially MPI transmission, gradually increased, far exceeding the time of floating-point calculation. In order to decrease the time proportion of MPI transmission, it is necessary to increase the calculation tasks undertaken by each core group without changing the amount of data transmitted by MPI. The hotspot function D-slash in lattice QCD application is a four-dimensional 9-point stencil operation, which is mentioned in Sect. 2. We reconstructed the data structure of the grid point data in the program and increased the lattice size responsible for each core group from 8^4 to 16^4 . According to the theory of stencil operations, the amount of calculations performed by each core group has increased by 16 times, and the proportion of data

at the edge of lattice blocks has been reduced by 10%. This method can complete a more accurate numerical simulation on the same scale.

From the calculation theory described in Sect. 2, we define the fermion fields as the following form (5), which can be represented in the computer as 24 floating-point numbers:

$$\begin{pmatrix} \alpha_1 + \beta_1 i & \alpha_2 + \beta_2 i & \alpha_3 + \beta_3 i & \alpha_4 + \beta_4 i \\ \gamma_1 + \delta_1 i & \gamma_2 + \delta_2 i & \gamma_3 + \delta_3 i & \gamma_4 + \delta_4 i \\ \mu_1 + \nu_1 i & \mu_2 + \nu_2 i & \mu_3 + \nu_3 i & \mu_4 + \nu_4 i \end{pmatrix} \quad (5)$$

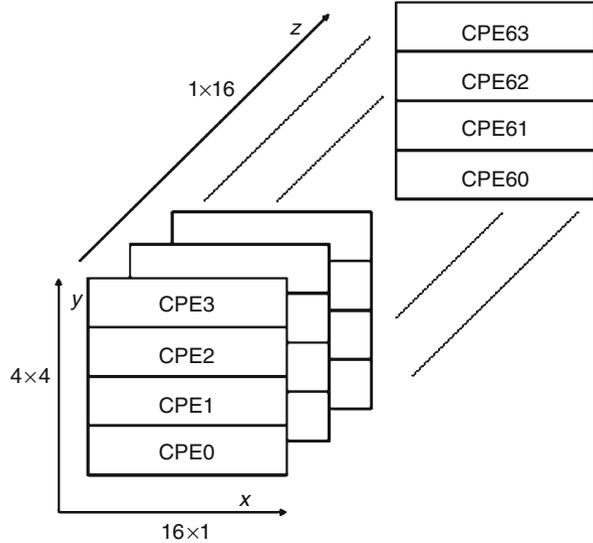
We define the gauge fields as the following form (6), which can be represented in the computer as 18 floating-point numbers:

$$\begin{pmatrix} a_1 + b_1 i & a_2 + b_2 i & a_3 + b_3 i \\ a_4 + b_4 i & a_5 + b_5 i & a_6 + b_6 i \\ a_7 + b_7 i & a_8 + b_8 i & a_9 + b_9 i \end{pmatrix} \quad (6)$$

If the computer stores data as single-precision floating-point numbers, then the size of the fermion field matrix on each grid point is 96B, and the size of the gauge field matrix in the positive direction of the four dimensions of each grid point totals $4 \times 72\text{B}$. That is, the total data size of the 8×8 lattice block is 24KB. SW26010 processor has independent 64KB LDM for each CPE, which is freely allocated by programmers and does not support cache coherence. Because of the limit of LDM, each CPE can only process the D-slash operation on at most 64 grid points with only one data transmission. Grid block size of 16^4 is unable to be stored in LDM of 64 CPEs of one core group at the same time, and the data for CPEs need to be redistributed.

To solve this problem, we have designed a method to distribute lattice point data to CPEs. Firstly, we divide the $16 \times 16 \times 16 \times 16$ four-dimensional grid block responsible for each core group into 16 three-dimensional grid cubes size of $16 \times 16 \times 16$ according to the t direction. In one iteration of D-slash calculation on a four-dimensional grid block, the problem was transformed into 16 D-slash calculations on three-dimensional grid cubes. Then, as shown in Fig. 5, the three-dimensional grid cube is sliced into 16 planes according to the z direction, and each plane is further sliced into 4 16×4 grid regions. Each CPE is responsible for the operation on one of the 16×4 grid regions in one D-slash calculation. The address of each two-dimensional grid region in the main memory is continuous and can be loaded into LDM through one DMA transmission easily.

Fig. 5 Data distribution of CPEs



3.2 Parallelization Method Based on Multiple Core Groups

The application supports parallelization on multiple core groups by a two-level “MPI+X” approach. As we mentioned in Sect. 3.1, each core group is responsible for the calculation of a lattice block size of 16^4 .

We cut a piece of four-dimensional space-time into multiple lattice blocks size of 16^4 . The task of each core group is the calculation of one of the lattice blocks. Each core group is bound to an MPI process managed by MPE, and the position of the grid block in the original four-dimensional space can be calculated based on the MPI process number, and the edge data of the grid block can be sent or received through MPI point-to-point communication. Threads on CPEs are generated by a high-performance lightweight thread library called *athread* and control tasks such as computation and memory access of CPEs. The entire lattice QCD application running process is shown in Fig. 6, which is mainly divided into the following steps:

- Step 1: Each MPE loads its assigned grid point data to the shared main memory, and each CPE loads its assigned grid point data to LDM according to its ID. Then initialize related parameters.
- Step 2: All the CPEs perform the D-slash calculation parallelly and update the data on every lattice. At this time, the algorithm needs the data of adjacent grid points. The adjacent grid points of the CPE’s edge grid points may not exist in LDM, and the edge grid points of CPEs or core groups need to be accessed. The data is transmitted through register communication between CPEs in a single core group. The edge data transmission between different core groups is managed by MPE using MPI communication, and CPEs use DMA to transmit these parts of data between shared memory and LDM.

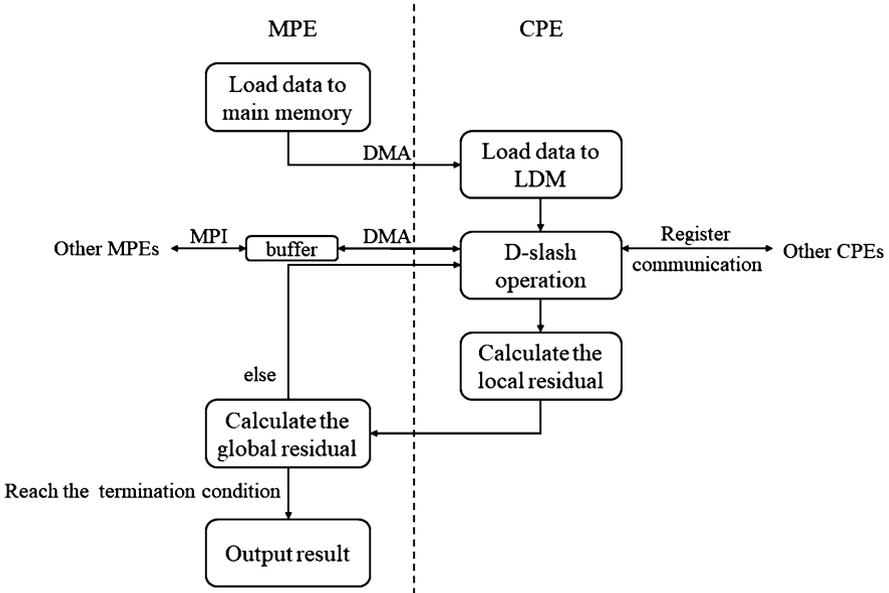


Fig. 6 Lattice QCD operation flow chart

Step 3: CPE calculate the residual value of the data in LDM and transfer the result to MPE. MPE performs global reduction through MPI communication (i.e., *MPI_All_reduce*) and calculates the global residual value.

Step 4: Determine whether the residual value meets the accuracy requirements. If it reaches, then all the CPEs will return the updated lattice data to MPE, and MPE will output the data to a file as final results. Otherwise, skip to Step 2 and continue the iteration.

Compared with the single core group parallelization method in [5], this multiple core group parallelization method changes the behavior of accessing edge grid point data during D-slash operation. When CPE accesses the grid point adjacent to the edge grid point, in addition to accessing the data in LDM of other CPEs through register communication, CPE may also need to access the data in the shared memory of other core groups, where MPI communication managed by MPE is necessary. In this way, the method of using multiple core groups can process larger-scale lattice data parallelly, and the distribution of lattice points in four dimensions can also be more flexible.

3.3 Algorithms and Pseudocodes

According to the characteristics of the SW26010 processor, we use CPEs to process the D-slash calculation and use *MPI_All_reduce* managed by MPE to calculate the residual value.

The D-slash algorithm pseudocode processing on CPEs is shown in Algorithm 1, which is the hotspot function of the program. The program loads the two-dimensional grid region *src* to LDM and initializes the updated grid point *result* in LDM. After that, a four-dimensional 9-point stencil operation is performed on *src* and update *result*. Then repeat this operation 16 times to finish the stencil operations. Finally, the program transfers *result* to the main memory as the calculation result.

Algorithm 1: D-slash

Variables:	<i>i</i>	One of the four dimensions x, y, z, and t
	<i>forward_i</i>	Adjacent grid points data in the positive direction of the i dimension
	<i>backward_i</i>	Adjacent grid points data in the negative direction of i dimension
	<i>src</i>	The data of the two-dimensional grid region responsible for this CPE
	<i>result</i>	Updated grid region data
Functions:	<i>init</i>	Initialize variables in LDM
	<i>dslash_spinor_local</i>	Perform D-slash calculation with input parameters
	<i>pull_data_from_memory</i>	Load data from memory to LDM by DMA
	<i>push_data_to_memory</i>	Store data from LDM to memory by DMA

```

BEGIN
  init(result)
  for j ← 0 to 15
  begin
    pull_data_from_memory(src)
    for i in (x, y, z, t)
    begin
      result += dslash_spinor_local(src, forward_i)
      result += dslash_spinor_local(src, backward_i)
    end
  end
  push_data_to_memory(result)
END
    
```

In the application implementation, grid points *forward_i* and *backward_i* needed by function *dslash_spinor_local* may not be stored in LDM, and data transmission is required. The address of the required data can be easily calculated by CPE ID. Adjacent grid points of the two-dimensional lattice region edges are stored in LDM of one of the other CPEs, which can be transmitted through register communication between CPEs. Adjacent grid points of the four-dimensional lattice block edges are stored in the main memory of one of the other core groups. These grid points can be transmitted by MPI communication managed by MPE and then transmitted to CPEs through stride mode DMA communication.

Algorithm 2: minimum residual solver

Variables:	<i>src</i>	All the lattice points in the whole problem
	<i>result</i>	Updated lattice points
	<i>residual</i>	The residual value, mentioned in Sect. 3.2. We set the accuracy requirement to 1.0×10^{-12}
Functions:	<i>dslash</i>	Perform D-slash operation on all lattice points
	<i>all_reduce</i>	Calculate the residual value of <i>result</i> by <i>MPI_All_reduce</i>
	<i>output</i>	Write <i>result</i> to file as output

```

BEGIN
  residual ← 1
  while residual ≥ 1.0 × 10-12
  begin
    dslash(src, result)
    residual ← all_reduce(result)
  end
  output(result)
END

```

The minimum residual solver of lattice QCD is shown in Algorithm 2. We control the entire program through MPE. After each iteration of D-slash calculation, all the MPEs in the system use *MPI_All_reduce* for global reduction and calculate the residual. This means MPI communication may be one of the performance bottlenecks of the entire program in large-scale parallelization.

4 Large-Scale Parallel Experiments and Analysis

4.1 Experiment Environments

The experiment uses Sunway TaihuLight supercomputer, and the specific software and hardware parameters and performance attributes are shown in Table 1. The result only includes the floating-point calculation part and ignores the part of I/O.

4.2 Experiments, Results, and Analysis

We test the D-slash algorithm and the minimum residual algorithm mentioned in Sect. 3.3. In the experiment, we use dynamically generated fermion field and gauge field input data. When testing the D-slash hotspot function, we make it loop 1000 times directly. There is a small amount of MPI data transmission between core groups, but MPI global reduction is not involved. When testing the minimum residual solver function, we make it loop until its global residual value is less than 1.0×10^{-12} . Compared with D-slash, the operation of MPI global reduction will have a certain impact on performance. The experiment results are shown in Fig. 7, where the gray dashed line is the progressive line. The D-slash function uses 1123200 cores and gets the performance of 63.703TFlops, and the minimum residual solver function achieved 139.147TFlops using 1347840 cores. The average performance on each core group is shown in Fig. 8. After the parallel scale of the two functions is expanded to one million cores, compared with the scale of 50,000 cores, they can maintain more than 44% of the performance.

Comparing the two functions in Figs. 7 and 8, although the minimum residual solver function has more MPI global reduction operations than D-slash, it can achieve higher performance. This may due to the small amount of floating-point calculation and the long data communication time in D-slash. The calculation of the global residual value in minimum residual solver involves a large number of floating-point calculations. Although the MPI global reduction is required, it can still effectively reduce the proportion of time wasted on data communication. Therefore, increasing the amount of calculation and reducing the proportion of data communication time are effective methods of optimizing lattice QCD applications on Sunway architecture.

Table 1 Experiment parameters and performance attributes

CPU	SW26010 1.45GHz, four core groups on one chip
Memory	8GB per core group/32GB per chip
Compiler	MPE: mpicc -O3 CPE: sw5cc -O3 Hybrid link: mpicc -O3
Results reported on the basis of	D-slash: calculation on CPEs Minimum residual solver: calculation on MPEs and CPEs
Precision reported	Single precision
System scale	Up to 1347840 cores (20736 core groups)
Measurement mechanism	Flop counts

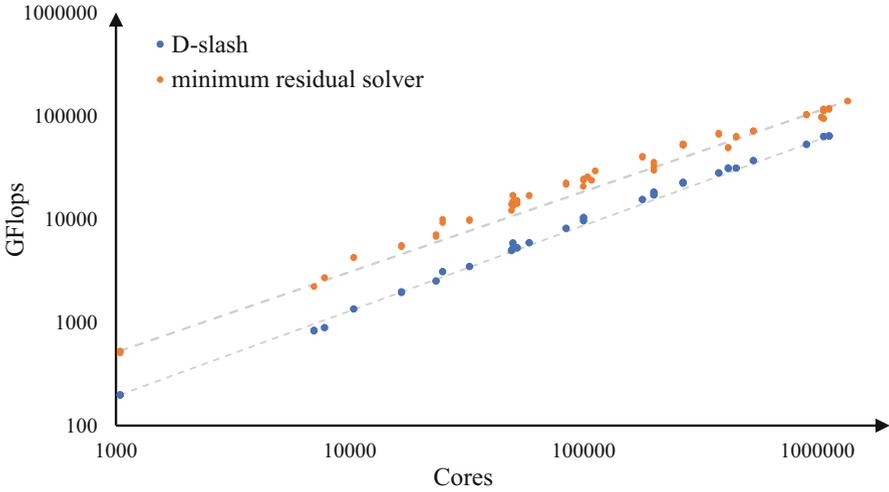


Fig. 7 Large-scale parallel performance results

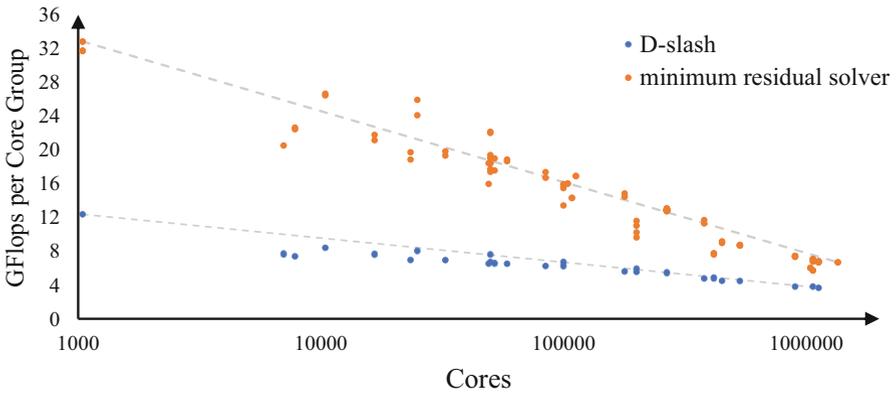


Fig. 8 Performance changes as the scale grows

5 Conclusions

We perform large-scale parallelization and performance optimization of lattice QCD applications based on SW26010 heterogeneous many-core processors. Through the analysis and improvement of the previous work, we propose a new grid point data distribution method which increases the amount of calculation on each core group and perform efficient parallel computing. The application achieved the maximum performance of 139.147TFlops using 1347840 cores on Sunway TaihuLight supercomputer and can maintain performance as the scale grows.

The lattice data distribution method proposed in this paper uses a large amount of register communication and DMA communication but does not conduct in-depth research on data transmission efficiency. We have found out that due to the parallel mode of “MPI+X,” all CPEs execute the same raw code and initiate DMA requests at almost the same time, which may cause bus congestion. In future research, we will conduct more research and optimization on hotspot functions of CPEs, try to avoid data bus contention during the program execution, and improve program performance.

Acknowledgments This research is supported by the National Key R&D Program (Grant No.2017YFB0203202) and the Natural Science Foundation of China (Grant No.11775229).

References

1. K.G. Wilson, Confinement of Quarks[J]. *Phys. Rev. D* **10**(8), 2445–2459 (1974)
2. P. Vranas, G. Bhanot, M.A. Blumrich et al., The BlueGene/L supercomputer and quantum ChromoDynamics[C], in *Conference on High Performance Computing (supercomputing)* (2006)
3. S.K. Gupta, P. Majumdar, Accelerating lattice QCD simulations with 2 flavors of staggered fermions on multiple GPUs using OpenACC-A first attempt[J]. *Comput. Phys. Commun.* **228**, 44–53 (2018)
4. I. Kanamori, H. Matsufuru, Practical implementation of lattice QCD simulation on SIMD machines with Intel AVX-512[J], in *International Conference on Computational Science and Its Applications* (2018), pp. 456–471
5. Z. Zhang, Z. Luan, C. Xu et al., Accelerating lattice QCD on Sunway many-core processor[C]. *Ubiquit. Comput.* 605–612 (2018)
6. J.L. Manferdelli, N.K. Govindaraju, C. Crall. Challenges and opportunities in many-core computing[J]. *Proc. IEEE* **96**(5), 808–815 (2008)
7. H. Fu, J. Liao, J. Yang et al., The Sunway TaihuLight supercomputer: System and applications[J]. *Sci. China Ser. F Inf. Sci.* **59**(7), 072001 (2016)
8. J. Zhang, C. Zhou, Y Wang et al., Extreme-scale phase field simulations of coarsening dynamics on the Sunway TaihuLight supercomputer[C], in *IEEE International Conference on High Performance Computing Data and Analytics* (2016)
9. H. Fu, C. He, B. Chen, et al., 18.9-Pflops nonlinear earthquake simulation on Sunway TaihuLight: enabling depiction of 18-Hz and 8-meter scenarios[C], in *IEEE International Conference on High Performance Computing Data and Analytics* (2017)
10. M. Creutz, Monte Carlo study of quantized SU(2) gauge theory[J]. *Phys. Rev. D* **21**(8), 2308–2315 (1980)
11. M. Creutz, Overrelaxation and Monte Carlo simulation.[J]. *Phys. Rev. D* **36**(2), 515–519 (1987)

Part III
Scientific Computing, Modeling and
Simulation

Reverse Threat Modeling: A Systematic Threat Identification Method for Deployed Vehicles



Mona Gierl, Reiner Kriesten, Peter Neugebauer, and Eric Sax

1 Introduction

The development of current driver assistance systems emerges from the intention to enhance the quality of mobility and to reduce the number as well as the severity of road accidents [1]. With the help of driver assistance features, vehicles increasingly take over various driving tasks (active braking, parking aid, lane keep assist, etc.) leading to connected and intelligent transport systems that require an equal integration of safety and security for reliable operation. Safety, with its ISO 26262 Standard [2], is already an integral part of the development life cycle, whereas automotive security is still an emerging domain that first gained relevance from opening up the automotive network to the outside world by using, for example, wireless interfaces and other interconnecting services. Since then, researchers have successfully demonstrated the negative impact of security exploits on safety, e.g., by disabling the brakes, killing the engine, etc. [3–5]. Compared to the statistical assessment of safety failure rates, successful attacks do not have accurate predictive empirical exploit rates. Instead, attacks depend on vulnerabilities that violate nonfunctional, mostly quantitative security goals (e.g., confidentiality, integrity, availability) and varying attacker techniques. This requires security to

M. Gierl (✉) · R. Kriesten · P. Neugebauer
Karlsruhe University of Applied Sciences, Institute of Energy Efficient Mobility, Bruchsal,
Germany
e-mail: gimo0002@hs-karlsruhe.de; krre0001@hs-karlsruhe.de; nepe0001@hs-karlsruhe.de

E. Sax
Karlsruhe Institute of Technology, Institute for Information Processing Technologies, Karlsruhe,
Germany
e-mail: eric.sax@kit.edu

regularly consider new unknown threats that might occur due to technological advancements or newly developed attacker methods.

Statistics of the Federal Motor Transport Authority in Germany show that the average age of present passenger cars is 9.5 years as of 2019 [6]—whereas the “age” is calculated based on the date of the car’s first approval. During this time, modifications and updates in hardware and software might occur due to maintenance, inspections, service, or occasionally unauthorized tuning. All these listed aspects can lead to new vulnerabilities that require further analysis to evaluate their potential impact. Additionally, new attack methods might be developed that were not considered beforehand. Hence, over the life cycle of a car, the question arises, whether integrated security measures are still appropriate or if new threats might disrupt its safety. Accordingly, a frequent security assessment after the vehicle is deployed appears reasonable.

Problem Statement: The attack surface of an approved vehicle is of dynamic nature due to occurring changes of hardware and software components and due to constantly evolving new attack methods. Exploiting a vulnerability might, in the worst case, impact road safety. Hence, a regular assessment of the attack surface (security assessment) is reasonable to identify and react to occurring threats. In this context, an assessment by a certification organization (as an accredited inspection authority, responsible for regular technical inspections) could be a conceivable occasion to assess the security of the vehicle as it is independent of the vehicle brand or any other competitive market conflicts of interest. Therefore, an analysis by a certification organization could help to build up additional trust in the vehicular security. However, independently assessing the security leads to a black box perspective, in which only external characteristics and driving features of the car are known. Thus, a threat modeling approach that extracts potential threats of a deployed car based on the few given system information is required.

Solution: In the context of this paper, a threat modeling approach for deployed vehicles is presented. A black box perspective is assumed, which allows to generate a high-level system model based on known driving features and their associated functionality. The generated system model serves as a basis for threat analysis. A systematic threat identification process is presented that considers functional similarities of available driving features between multiple vehicles. The graphical representation of an attack tree is chosen to structure the identified threats and their attack paths.

Contribution: This paper serves as a first step to address security analyses after the deployment of a vehicle. Based on available threat and risk analyses commonly applied in the concept phase, a threat modeling method for deployed vehicles is introduced. Results of the analysis are expected to identify potential high-risk threats and shall serve as a feasible input for future security assessment approaches to ensure the road safety of approved vehicles.

2 Background

Automotive security and safety are interacting disciplines. A successful attack with a safety-related impact implies an unjustifiable high risk to the driver and the surrounding traffic. Therefore, security by design is a promising attempt in which security analyses during the early design phase help to identify potential threats and to derive appropriate security measures [7]. In this context, the SAE J3061 Cybersecurity guidebook for cyber-physical vehicle systems [8] serves as a guideline since 2016. Furthermore, an official automotive security standard (ISO/SAE 21434) is currently under development, whereas the official release is planned for the end of 2020 [9]. The guidelines demand the usage of threat analysis and risk assessment (TARA) methods in the early concept phase of vehicle development.

2.1 Threat Analysis and Risk Assessment (TARA)

According to the SAE J3061 [8], a TARA is a theoretical security analysis applied during the concept phase that aims to identify potential threats and assesses the associated risk. Various TARA methods exist that follow threat identification, threat classification, and risk analysis schemes but differ depending on their system knowledge and applicability for the automotive domain.

In case of threat assessment based on a deployed vehicle, even fewer system information are provided since most technological implementations are confidential. Thus, the deployed car can be described as a black box that has specific driver assistance systems or driving features. Many fundamental driving features (e.g., antilock braking system (ABS), electronic stability program (ESP)) are equally represented in various cars, whereas the physical characteristics of the features do not change. Thus, finding threats that are solely associated with driving features or functionality could help to find comparable and reusable analysis results over multiple vehicles with similar driving features. Still, an attacker's entry point depends on the vehicular interfaces and thus on its electronic architecture. Therefore, the system model has to include both, a functional and an electronic representation, whereas the electronic representation is only known on a very high level, including, for example, interfaces.

Overall, the challenge for a security assessment of a deployed car revolves around the threat identification process that differs in terms of given system information. Whenever threats are identified, known classification and ranking schemes can be applied to assess the associated risk. Reusing known classification schemes also allows the results to be comparable.

3 Related Work

As of today, there are several TARA methods aiming to equally integrate safety and security and designed for the concept phase of a vehicle (white box approaches). A systematic literature survey on safety and security co-analyses is conducted by Lisova et al. [10], a survey on integrated safety and security risk assessment methods is given by Chockalingam et al. [11], and a review especially for automotive TARA methods is conducted by Macher et al. [12]. Additionally, Ma et al. [13] address threat modeling techniques for automotive security analysis during the development life cycle.

However, to the best of our knowledge, there are currently no threat modeling approaches following a black box approach that regularly monitor the vehicular security after the market introduction. In fact, Lisova et al. [10] states that the known safety and security co-analyses lack the evaluation of their update handling support. In other words, it is not clear to what extent the available co-analyses are able to consider dynamic security changes due to system updates. An evaluation to analyze the applicability of current threat modeling approaches in the context of a deployed car is required. Thus, the question arises to what extent the methods are applicable in terms of a black box threat modeling approach. The following section briefly lists the automotive TARA methods and discusses their applicability in the context of this work.

3.1 TARA Methods

The SAE J3061 provides a list of known TARA methods in its appendix including EVITA, TVRA, OCTAVE, HEAVENS, Attack Trees, and Software Vulnerability Analysis Overview [8]. Further threat analysis techniques are represented in the automotive research field. A review of these methods is provided by Macher et al. [12] including FMVEA, SAHARA, SHIELD, CHASSIS, BDMP, Threat Matrix, and Binary Risk Analysis (BRA) which evaluates the methods in terms of applicability for the concept phase of automotive systems. Results of [12] show the TARA characteristics and determine EVITA, Attack Trees, HEAVENS, FMVEA, SAHARA, SHIELD, CHASSIS, and BDMP to be applicable in the automotive context. A brief overview of these TARA methods is given in Table 1.

The presented set of existing TARA methods (in Table 1) shall serve as a toolbox to develop a threat modeling approach for deployed vehicles. For threat modeling in general, a system model, a threat identification procedure, and a way to structure the identified threats are of relevance. In case of organizing known threats, both BDMP and attack trees facilitate a structuring technique to graphically document attack steps and their interdependencies. Since the current paper focuses on the security of deployed cars, a safety analysis previous to a security analysis is out of scope for this work; thus, in Sect. 4, attack trees are chosen over BDMP. Detailed

Table 1 List of automotive TARA methods

Method		Description
EVITA	[14]	The E-safety Vehicle Intrusion Protected Applications (EVITA) project uses so-called dark-side scenarios in combination with attack trees to identify potential threats. The project defines a domain-based network architecture as its system under investigation. According to EVITA, the threats are identified largely by experience and structured within attack trees [14]
Attack Trees	[15]	Attack trees are derived from the safety equivalent of fault trees and are a graphical representation of possible attack steps that lead to a successful exploitation of the associated system. The root node consists of the malicious attack goal, whereas the intermediate nodes are attack steps or sub-goals that are required to trigger the main attack goal. Overall, attack trees provide a technique to structure identified attack steps graphically
HEAVENS	[16]	The Healing Vulnerabilities to Enhance Software Security and Safety (HEAVENS) framework is based on Microsoft's Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege (STRIDE) classification scheme for threat analysis. STRIDE is a threat model which aims to identify threats by providing threat categories. Additionally, HEAVENS developed a risk ranking method that determines the Security Level (SL) of a threat based on the identified threat and impact level
FMVEA	[17]	FMVEA is an extension of the Failure Mode and Effect Analysis (FMEA) to additionally address security vulnerabilities. FMVEA focuses on failure and threat modes of components. To identify the relevant components, the approach decomposes its underlying system into functions and later maps them to their according devices. Data flow diagrams are used to analyze the effect on the system, the failure and threat mode, and their respective causes
SAHARA	[18]	The Security-Aware Hazard and Risk Analysis (SAHARA) combines the outcome of a security analysis with a Hazard and Risk Analysis (HARA). Both analyses are conducted separately following known schemes of STRIDE and HARA described by the ISO 26262. Identified threats which are classified to impact the safety of a vehicle are selected to undergo further safety analysis
SHIELD	[19]	The SHIELD approach assesses the security, privacy, and dependability (SPD) of embedded systems. The approach is based on the Common Criteria classification scheme to quantify the SPD level
CHASSIS	[20]	The Combined Harm Assessment of Safety and Security for Information Systems (CHASSIS) approach models misuse cases with the help of Unified Modeling Language (UML)-diagrams. The UML-diagrams are analyzed by using guide words from the Hazard and Operability Study (HAZOP) and on basis of security flaws (structured according to their Confidentiality, Integrity, and Availability)
BDMP	[21]	The Boolean Logic Driven Markov Processes (BDMP) is an extension of the fault tree that includes Markov models [22]. Later, the security-informed BDMP was introduced as an extension of the fault tree to include attack steps as it is known from attack trees [21]

information on how to generate attack trees in the automotive context is provided by the EVITA project. Further, threat identification methods are required to generate comprehensive attack trees. In Table 1, mostly Microsoft's STRIDE is mentioned in case of threat identification. To apply STRIDE, a detailed system model is required for a thorough threat identification process.

FMVEA and CHASSIS provide explicit details about system modeling. CHASSIS is applied during requirement engineering to identify possible misuse cases from known system use cases. As requirement engineering does not have detailed information about future system implementation, it has similarities with the assumed black box approach of this work. Accordingly, in the proposed approach, misuse cases are derived from known use cases or driving features of a deployed car. To extract as many information for the system model as possible, several tools are available to extract vehicle-specific information about integrated ECUs and other components including diagnostic testers or port scanners that are able to scan the vehicular network for available communication components as presented in [5]. As soon as more system-specific information is available, threats to components are of interest. In this context, FMVEA gives the guiding idea to differentiate between the system functionalities and the available hardware components and to extract threat modes based on the physical and functional system overview to make the analysis re-applicable over multiple vehicles.

4 Approach

In this section, a threat modeling method for deployed vehicles is presented. It is designed to enable threat assessment from the perspective of a certification organization. An overview of the stepwise procedure of the presented approach is given in Fig. 1.

In general, threat analysis involves (a) the modeling of the system and (b) the theoretical prediction of the attacker's steps to identify threats and their attack vectors that can later be ranked based on existing risk assessment methods. A successful attack consists of various interconnected attack steps that lead to the exploitation of the system. To represent the relations between the system and the attack steps, the use of attack trees as a graphical representation of potential threats is suggested. Hence, the approach aims to develop attack trees that graphically list all potential threats, which might provoke a safety-related attack goal.

As a first step, the root node of the attack tree is determined by defining a misuse case that might impact the safety of the car. Later (in Sect. 5), unintentional braking of the car is chosen as an exemplary misuse case. The second step originates from the following question: Which driving features might trigger the defined misuse case? Driving features include all driving-related functions that influence the driving behavior of the car, e.g., implemented driver assistance systems. Hence, a list of relevant driving features is created in step 2. To further design the system model, the logical data flow between the sensors and actuators of the listed driving features

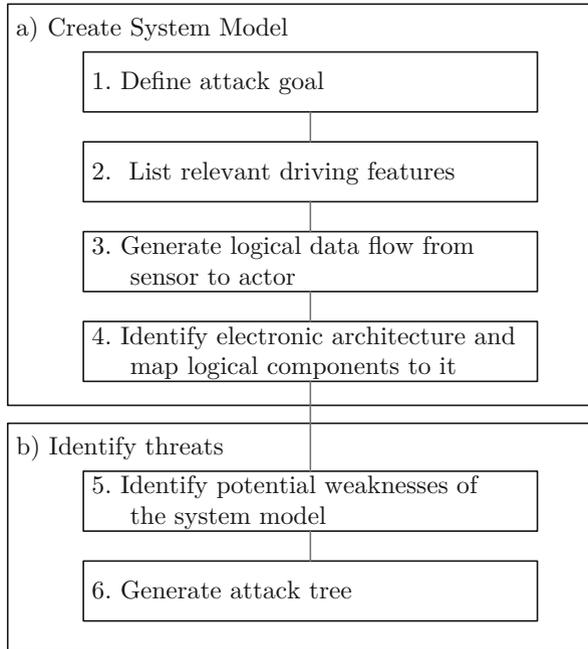


Fig. 1 Proposed threat modeling procedure for a deployed vehicle. First, a system model is generated from the provided system knowledge. Based on the system model, threats are identified

has to be defined. The data flow is typically known in the development phase; however, it is not known in a black box approach considering the perspective of certification organizations. Nevertheless, with some technological background, a high-level functional representation of the data flow can be recreated by analyzing the functionality of the associated driving features.

To depict the functional system model of the data flow, a data flow diagram is chosen. The typical structure of the functional system model is given in Fig. 2, in which the arrows represent the data flow from sensor inputs to actuator output. An example with more details is given in Sect. 5. Each driving feature represents a functionality of the car and therefore exists in the electronic architecture (e.g., as software code on an electronic control unit (ECU)). Every associated ECU running the software implementation has different communication interfaces that might serve as an attacker's entry point. Hence, also shown in Fig. 2, mapping between the driving features and the associated electronic hardware finalizes the system model in step 4 of the approach. As an example, the ABS driving feature is expected to run on the ABS ECU from the electronic architecture. To extract the associated ECUs, methods that obtain system or implementation information from the deployed car should be used as much as possible (e.g., diagnostic testers or port scanners as presented in [5]). In the context of this work, an exemplary conceptual reference architecture of a vehicle is used as it is presented in [23] (Fig. 3).

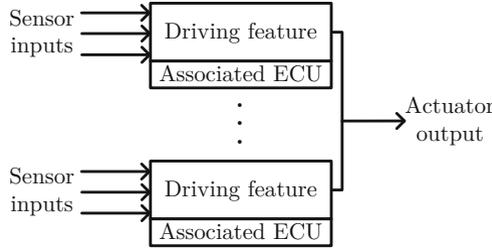


Fig. 2 General structure of the generated functional overview. Arrows represent the logical data flow from sensor to actuator. In between, driving features are listed. Each driving feature represents a functionality of the car which is implemented on an associated ECU

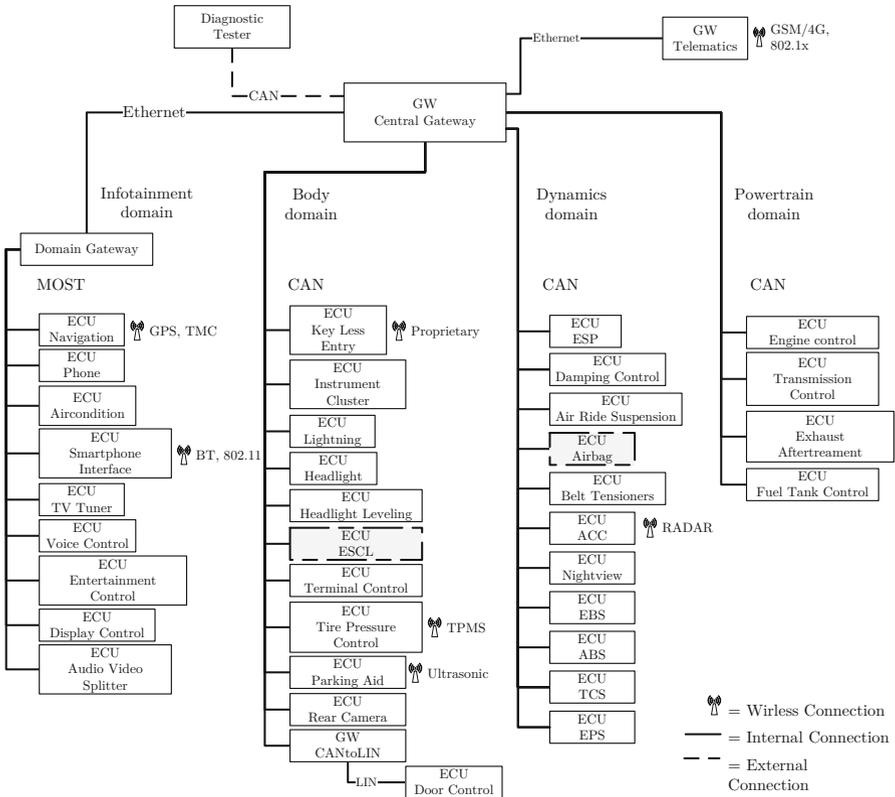


Fig. 3 Common conceptual reference architecture of a car according to [23]

On basis of the designed system model, the potential threats can be identified. A structured procedure to identify relevant threats is required. Thus, before the actual attack tree is designed, an analysis of the modeled system is suggested in step 5. Each logical connection and component from the functional system model

(Fig. 2) is examined for potential weaknesses (e.g., the manipulation of a software implementation). Accordingly, following this approach, patterns of threats that occur over a range of cars are expected to be found since the logical context of the driver assistance systems are similar over different vehicle brands. For example, on a high abstraction level, functional threat patterns can be seen by either manipulating the input and the functional implementation or sending an unauthorized message that triggers the output, therefore leading to the next attack step. Based on the results, a structured attack tree can be created where the attack steps are derived from the previously listed potential weaknesses. Further down in the attack tree, the electronic architecture is of relevance to identify the attacker's entry points. Also, if security measures are known to the analyst, the measures should be listed, so that they can be considered during the attack tree generation. Finally, the attack tree is generated from the root node downward, always considering the question which attack steps might trigger the overall attack goal or its intermediate steps. Finally, published or known attacks as provided by Sommer et al. [24] should be used to support the thorough attack tree generation.

5 Example

The following example is designed to serve as a guideline to enable future threat modeling based on the presented approach for deployed vehicles. Therefore, the following section is enumerated based on the required steps. Further, the guideline strives to be reusable and applicable over a range of current common cars, which is why no specific vehicle type is determined. Instead, the conceptual reference architecture of Fig. 3 is used:

- (1) *Define safety-relevant attack goal:* The presented threat modeling in the operation phase of the vehicle (from deployment to decommission) aims to address safety-related threats that might occur due to component changes or evolving attack methods. Especially misuse cases that might impact safety-relevant driving maneuvers are of interest. A basic selection of safety-relevant maneuvers of a vehicle includes braking, steering, and accelerating. Per scenario, either unintentional execution or loss of functionality might impact the road safety. For the subsequent guideline, the unintentional braking scenario is chosen as an exemplary attack goal to demonstrate the presented approach.
- (2) *List relevant driving features:* As mentioned in Sect. 3, the threat identification process of the available TARA methods varies depending on the provided system knowledge. Therefore, information gathering is a crucial step for a thorough security analysis. As from the perspective of a deployed vehicle, there is only few information about the integrated technology. Therefore, driving features that might influence chosen misuse case are the only reasonable starting point. Hence, as a second step, a list of available driving features that might interact with a previously defined attack goal (e.g., unintentional braking) is

Table 2 List of driving features that are able to influence the braking behavior

Driving features which might impact the brake behavior:	
Electronic brake system (EBS)	Active cruise control (ACC)
Antilock braking system (ABS)	Active brake assist (ABA)
Traction control system (TCS)	Park assist (PA)
Electronic stability program (ESP)	

created. This list is shown in Table 2. Information about interfaces and ECUs are in some cases provided by the original equipment manufacturers (OEM). Additional scanning tools might be used to extract the ECUs over on-board diagnostics (OBD) as described by Ring et al. [5].

- (3) *Generate logical data flow*: On basis of the gathered information, the system model can be created. As a first step, the system functionalities and data flow from sensors to actuators are derived from the known driving features. Even without exact system information, a high-level data flow can be derived from Table 2, since the functionalities of the car are always based on the same physical fundamentals. An exemplary functional overview showing the data flow of driving features that are able to unintentionally trigger the brake is shown in Fig. 4. The graphic also includes a mapping of the functional driving features to the respective ECUs of the conceptual reference architecture and potential attack points A1–A5, which are addressed and explained in step 5 of the example.
- (4) *Identify electronic architecture and map logical components*: To extract available electronic hardware components, methods that obtain system or implementation information from the deployed car should be used as much as possible. In the context of this work, the exemplary conceptual reference architecture from Fig. 3 is used. A mapping between the driving features from Table 2 and the reference architecture allows to reduce the conceptual architecture from Fig. 3, to show only core components of the architecture that might trigger the exemplary unintentional attack scenario. The reduced reference architecture is shown in Fig. 5 and serves as a basis for the subsequent threat identification process.
- (5) *Identify potential weaknesses*: In this step, the functional system model (Fig. 4) is analyzed to find potential weaknesses of the data flow. Potential weaknesses are identified by analyzing the functional system model with the following question in mind: What change to the functional system could trigger an unintentional brake scenario? The results aim to serve as a starting point for later threat identification. In Fig. 4, the attack points (A1–A5) mark the potential weaknesses in the logical system model. Overall, the example exposes three methods that might trigger the unintentional brake scenario:
 - Violate integrity of the input values of the ECU by either manipulating the environment or spoofing data on the communication bus (A1, A3 in Fig. 4)

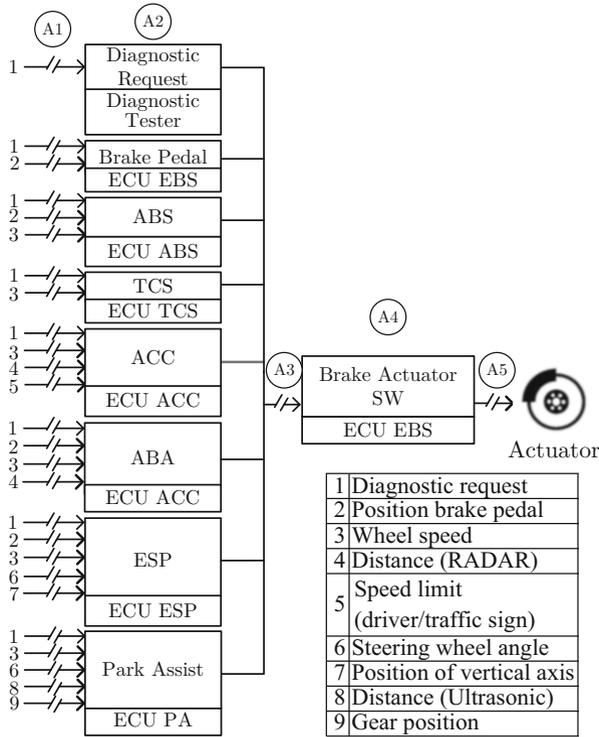
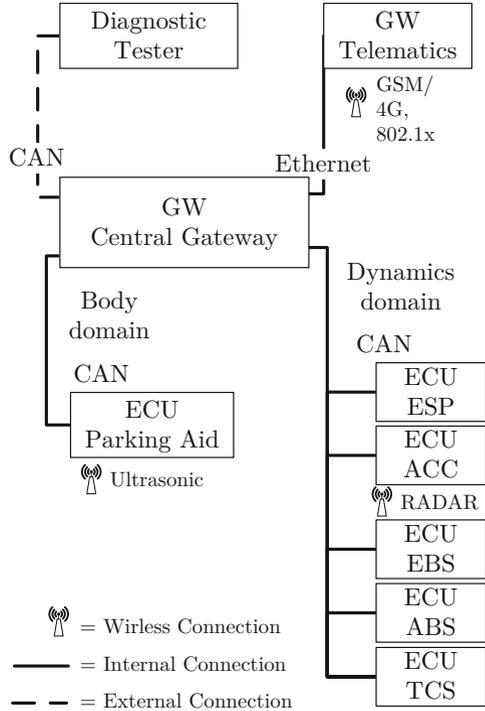


Fig. 4 Attack points in the functional system overview. The functional system overview is structured as in Fig. 2; further, possible attack points (A1–A5) are marked as interruptions in the data flow

- Compromise the ECUs of the vehicle, e.g., by reflashing the software (A2, A4 in Fig. 4)
 - Data spoofing to trigger the brake directly (A5 in Fig. 4)
- (6) *Generate attack tree*: The previous steps gathered all the required information to generate the attack tree. Starting from the top, the root node describes an attack goal or a misuse case (unintentional triggering of the brake). Each node below represents an intermediate attack step or sub-goal. The intermediate attack steps are derived from the identified potential weaknesses from step 5. This is illustrated by the first instance of the attack tree, which outlines the aforementioned three methods A3–A5 in Fig. 6. In an iterative manner, further potential threats are identified by answering the following question: Which changes might trigger the respective sub-goal? An excerpt of the exemplary attack tree is given in Fig. 6. For the sake of visibility and due to limited space, the attack tree consists of sub-trees marked by dashed lines and double-framed nodes. The sub-trees with dashed lines identify potential threats that might occur because of compromised driving features. These sub-trees were solely

Fig. 5 Reduced reference architecture listing electronic components potentially involved in an unintentional brake scenario



introduced to reduce the complexity of the main tree. On the other hand, the double-framed nodes describe sub-trees that require the architecture to further differentiate the required attack steps. Thus, these sub-trees are architecture-dependent meaning that the content of these sub-trees differs according to the components inside a vehicle. An exemplary architecture-dependent attack tree is shown in Fig. 7. As can be seen, the steps to achieve the sub-goal “Gain access to the vehicle network” vary depending on the underlying network architecture and integrated components. For example, to gain access remotely, the interfaces of the telematic gateway are possible entry points; however, to send a manipulated message onto the internal CAN bus of the dynamic domain, the central gateway has to forward the injected messages. Because of varying interface types and components in the network structure, an architecture-dependent analysis of potential threats has to be conducted for each vehicle.

The attack tree is finalized after each sub-tree reaches its initial attack step or entry point. The results then show the potential threats to trigger an unintentional brake event and all the required attack paths. As stated before, the results of the proposed approach list potential threats of a deployed vehicle and serve as a first step toward threat analysis of a deployed vehicle. Certainly, further discussions on threat classification and risk analysis are required. Still, the information about potential

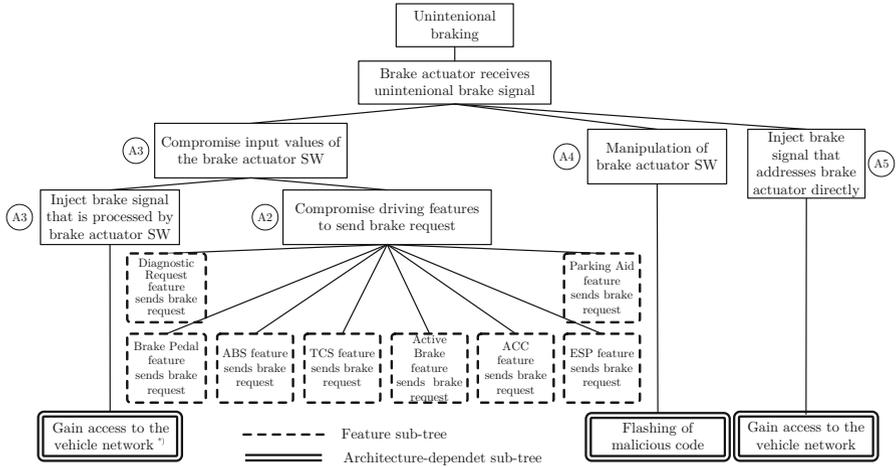


Fig. 6 Generated attack tree of the unintentional braking scenario

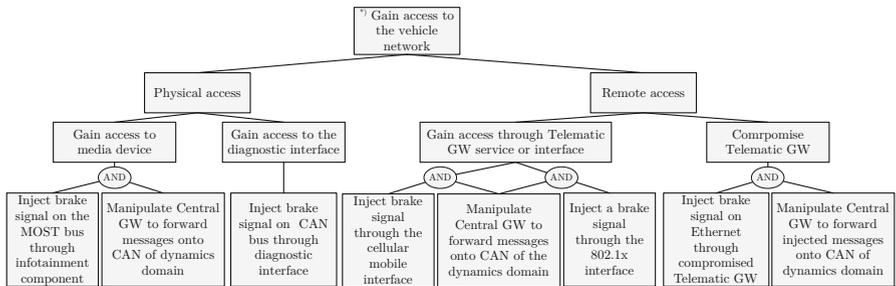


Fig. 7 Generated architecture-dependent sub-tree to gain access to the vehicle network. Attack steps are generated with the reference architecture in mind

threats should be considered in future security assessments during the operational phase of the vehicle.

6 Conclusion

In the context of this paper, a systematic threat modeling approach for deployed vehicles is introduced. The proposed threat identification process is a first step toward threat analysis of deployed vehicles and aims to identify new threats that might occur because of system changes during the operation of the car. As security might impact the road safety, a regular re-evaluation of the system security over the whole vehicle life cycle is reasonable. In this case, an independent analysis from a certification organization could help to build up additional trust in the vehicular

security. Thus, a black box perspective is assumed, with which a high-level system model is generated. The system model is based on known driving features and their associated functionality and serves as a basis for threat analysis. A systematic threat identification process is presented that considers functional similarities of the driving features between multiple vehicles. The graphical representation of an attack tree is chosen to structure the identified threats and their attack paths.

A special focus lies on the analysis of driving features. Separating the driving functionality from the physical electronic architecture allows to subdivide the attack tree into functional-dependent attack steps and architecture-dependent attack steps, which in combination might threaten the safety of the car. The results of the analysis aim to show potential safety-relevant threats of the system that are detected either because of new, previously unknown vulnerabilities or because changes to the system unknowingly widened the attack surface. Additionally, the results shall serve as a feasible input for future security assessment approaches to ensure the road safety of approved vehicles. In this context, further research is required. Therefore, future work will concentrate on identifying possible security assessment techniques for the deployed vehicle and will focus on deriving possible practical security testing methods.

Acknowledgments The presented work was funded by GTÜ Gesellschaft für technische Überwachung mbH in Stuttgart, Germany.

References

1. K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, H. Winner, Three decades of driver assistance systems: Review and future perspectives. *IEEE Intell. Transp. Syst. Mag.* **6**(4), 6–22 (2014). <https://doi.org/10.1109/MITS.2014.2336271>
2. ISO 26262-2018: Road vehicles – Functional Safety
3. C. Valasek, C. Miller, A Survey of Remote Automotive Attack Surfaces (2014). https://ioactive.com/pdfs/IOActive_Remote_Attack_Surfaces.pdf
4. K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, Experimental security analysis of a modern automobile, in *IEEE Symposium on Security and Privacy (SP)* (2010), pp. 447–462. <https://doi.org/10.1109/SP.2010.34>
5. M. Ring, J. Dürrwang, F. Sommer, R. Kriesten, Survey on vehicular attacks – building a vulnerability database, in *2015 IEEE International Conference on Vehicular Electronics and Safety (ICVES)* (2015), pp. 208–212. <https://doi.org/10.1109/ICVES.2015.7396919>
6. Kraftfahrt-Bundesamt: Steigendes Durchschnittsalter bei den Personenkraftwagen (2019). https://www.kba.de/DE/Statistik/Fahrzeuge/Bestand/Fahrzeugaalter/fahrzeugaalter_node.html
7. ENISA: Good practices for security of smart cars. <https://www.enisa.europa.eu/publications/enisa-good-practices-for-security-of-smart-cars>
8. SAE International: Cybersecurity Guidebook for Cyber-Physical Vehicle Systems (Jan 2016)
9. ISO/SAE DIS 21434: Road Vehicles – Cybersecurity engineering
10. E. Lisova, I. Sljivo, A. Causevic, Safety and security co-analyses: A systematic literature review, in *2019 IEEE 43rd Annual Computer*, p. 833. <https://doi.org/10.1109/COMPSAC.2019.00122>

11. S. Chockalingam, D. Hadžiosmanović, W. Pieters, A. Teixeira, P. van Gelder, Integrated safety and security risk assessment methods: A survey of key characteristics and applications, in G. Havarneanu, R. Setola, H. Nassopoulos, S. Wolthusen (eds.) *Critical Information Infrastructures Security Lecture Notes in Computer Science*, vol. 10242 (Springer, Cham, 2017), pp. 50–62. https://doi.org/10.1007/978-3-319-71368-7_5
12. G. Macher, E. Armengaud, E. Brenner, C. Kreiner, A review of threat analysis and risk assessment methods in the automotive context, in A. Skavhaug, J. Guiochet, F. Bitsch (eds.) *Computer Safety, Reliability, and Security*, Lecture Notes in Computer Science, vol. 9922 (Springer, Cham, 2016), pp. 130–141. https://doi.org/10.1007/978-3-319-45477-1_11
13. Z. Ma, C. Schmittner, Threat Modeling for Automotive Security Analysis. *Advanced Science and Technology Letters*, pp. 333–339. Science & Engineering Research Support soCietY (2016). <https://doi.org/10.14257/astl.2016.139.68>
14. O. Henniger, L. Aprville, A. Fuchs, Y. Roudier, A. Ruddle, B. Weyl, Security requirements for automotive on-board networks, in *9th International Conference on Intelligent Transport Systems Telecommunications (ITST)* (2009), pp. 641–646. <https://doi.org/10.1109/ITST.2009.5399279>
15. B. Schneier, *Secrets and Lies: Digital Security in a Networked World* (Wiley, Indianapolis, Ind. 2013)
16. M. Islam, C. Sandberg, A. Bokesand, T.O., H. Broberg, P. Kleberger, A. Lautenbach, A. Hansson, A. Söderberg-Rivkin, S.P. Kadhivelan: HEAVENS: Security models: HEALing vulnerabilities to ENhance software security and safety. <https://autosec.se/holisecc-results/>
17. C. Schmittner, Z. Ma, P. Smith, FMVEA for safety and security analysis of intelligent and cooperative vehicles, in *Computer Safety, Reliability, and Security*, Lecture Notes in Computer Science, vol. 8696 (Springer, Cham, 2014), pp. 282–288. https://doi.org/10.1007/978-3-319-10557-4_31
18. G. Macher, H. Sporer, R. Berlach, E. Armengaud, C. Kreiner, SAHARA: A security-aware hazard and risk analysis method, in *2015 Design, Automation & Test in Europe Conference & Exhibition* (2015) Piscataway. IEEE, pp. 621–624. <https://doi.org/10.7873/DATE.2015.0622>
19. A. Fiaschetti, V. Suraci, F.D. Priscoli, The SHIELD framework: How to control security, privacy and dependability in complex systems, in *2012 Complexity in Engineering COMPENG*, pp. 1–4. <https://doi.org/10.1109/CompEng.2012.6242962>
20. C. Raspotnig, P. Karpati, V. Katta, A combined process for elicitation and analysis of safety and security requirements, in *Enterprise, Business-Process and Information Systems Modeling*, Lecture Notes in Business Information Processing, vol. 113 (Springer, Berlin/Heidelberg, 2012), pp. 347–361. https://doi.org/10.1007/978-3-642-31072-0_24
21. L. Pietre-Cambacedes, M. Bouissou, Modeling safety and security interdependencies with BDMP (Boolean logic Driven Markov Processes), in *2010 IEEE International Conference*, pp. 2852–2861. <https://doi.org/10.1109/ICSMC.2010.5641922>
22. M. Bouissou, J.L. Bon, A new formalism that combines advantages of fault-trees and Markov models: Boolean logic driven Markov processes. *Reliab. Eng. Syst. Saf.* **82**(2), 149–163 (2003). [https://doi.org/10.1016/S0951-8320\(03\)00143-1](https://doi.org/10.1016/S0951-8320(03)00143-1)
23. K. Beckers, J. Dürrwang, D. Holling, Standard compliant hazard and threat analysis for the automotive domain. *Information* **7**(3), 36 (2016). <https://doi.org/10.3390/info7030036>
24. F. Sommer, J. Dürrwang, R. Kriesten, Survey and classification of automotive security attacks. *Information* **10**(4), 148 (2019). <https://doi.org/10.3390/info10040148>

PRNG-Broker: A High-Performance Broker to Supply Parallel Streams of Pseudorandom Numbers for Large-Scale Simulations



Andre Pereira and Alberto Proenca

1 Introduction

Scientific data analyses are developed to test, validate, and simulate hypothesis, theories, and phenomena. These applications often rely on tasks that operate on large sets of measured data with an associated measurement uncertainty and that may be computationally intensive. A common strategy to limit this uncertainty is to sample the data within its margin of error, often resorting to Monte Carlo algorithms, which may account for a significant portion of the overall execution time. This need for randomness in a deterministic environment created the demand for algorithms that provide seemingly random numbers.

Pseudorandom number generation (PRNG), the process of generating apparently random numbers on digital chips, is a well-studied topic, with the first computer-based algorithms being suggested as early as 1951 [24]. There are several PRNGs available with excellent statistical quality, as well as implementations on various programming environments with reasonable performance. However, the generator performance is often overlooked by noncomputer scientists, which may lead to a significant application performance degradation.

Three key issues should be considered when selecting a PRNG for a scientific application that requires very large sets of PRNs: the statistical quality, which is out of the scope in this work, the computational performance of the algorithm and its

This work has been supported by FCT (Fundacao para a Ciencia e Tecnologia) within the R&D Units Project Scope: UIDB/00319/2020.

A. Pereira (✉) · A. Proenca

Algoritmi Centre, Department of Computer Science, Universidade do Minho, Gualtar, Portugal
e-mail: ampereira@di.uminho.pt; aproenca@di.uminho.pt

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_14

167

implementation, and how an implementation is used in the code. These issues are particularly critical in parallel environments, such as in multicore and many-core compute servers, where algorithmic and computational inefficiencies may lead to significant performance degradation and poor scalability.

This paper presents PRNG-Broker, a *middleware* that manages the interaction of the user code with efficient implementations of popular PRNGs provided by widely adopted libraries in the scientific community. It transparently implements a dual-buffer approach for efficient management of large sets of PRNs, which can be natively generated in multicore and many-core servers, as well as offloaded to hardware accelerators and other servers. This paper also evaluates different approaches to use PRNGs in these libraries and their performance on different devices, detailing their PRN throughput and possible memory transfer costs associated with distributed memory environments. The analysis used as case studies three real parallel applications related to the search of the Higgs boson [2], which required different amounts of PRNs.

The PRNG-Broker currently supports different implementations of the popular Mersenne Twister PRNG [16], provided by the ROOT [19], MKL [25], STL [23], and cuRAND [17] libraries, but this list can be easily extended in future releases. A PRNG from the permuted congruential generator (PCG) family [18] was also evaluated, as the authors claim that it performs better than any other algorithm; however, it is not yet fully accepted by the scientific community.

This paper is structured as follows: Sect. 2 contextualizes the generation of random numbers, presenting the most popular PRNGs, the distribution transformations, and the different approaches to use them in parallel multicore/many-core servers with accelerator devices; Sect. 3 provides an in-depth analysis of the PRNG-Broker library; Sect. 4 presents the three case studies used to evaluate the different PRNGs and their implementations; Sect. 5 evaluates the different PRNG implementations in the case studies; and Sect. 6 makes a critical analysis of the developed work with suggestions for further improvements.

2 Random Number Generation

Random numbers are used in a wide spectrum of applications where unpredictability is required, including statistical data sampling, scientific computing, gaming, and cryptography. Different applications often require specific properties from random numbers, for which different random number generators may be used. In this context, these can be broadly classified as true random number generators (TRNGs) or pseudorandom number generators (PRNGs).

TRNGs are based in physical random processes to generate random bit strings. The most common example of a TRNG is the coin toss of a symmetrical coin, where one can expect either heads or tails with a 50% certainty. There are no correlations among generated numbers, but these generators are usually slow, not

suit for large-scale computing, and their results cannot be replicated, which makes debugging code harder.

PRNGs attempt to approximate the properties of truly random numbers, such as no repetition of sequence of values for a long period and no correlation between generated numbers. However, the generated values are not truly random as they are determined by an initial value (seed, which ensures result reproducibility). The main benefit of this type of random generator is their performance, which, depending on the algorithm, may scale with the increase of available cores. This type of generators is mostly used in scientific applications due to its higher performance and adequate mathematical properties. A short introduction to the most popular PRNGs, distribution transformations, and libraries follows through the next subsections.

2.1 Popular PRNG Algorithms

A wide range of algorithms to generate PRNs is currently available, each with strengths and weaknesses that may make them best suited for different uses. The quality of a PRNG randomness is usually evaluated by a set of benchmarks, such as the DieHard [14] and TestU01 [11] suites.

The scientific community has been using several PRNG algorithms, but one stands above all other in popularity: the Mersenne Twister [16]. This algorithm was developed in 1997 and features a period of $2^{19937} - 1$, passes most statistical tests, and is extremely fast to generate both 32- and 64-bit numbers. It has some limitations, such as low throughput, but these are often overcome by alternative implementations of this algorithm, which take advantage of vector/SIMD instructions, GPU architectures, and multithreaded environments.

Recently, the PCG family of PRNGs was proposed [18], claiming better statistical quality and computational performance, for both single-threaded and multi-threaded environments. Even though it is not yet fully accepted by the scientific community, the PCG RXS-M-XS 64 generator (a linear congruential generator (LCG)) will also be included in our performance evaluation in Sect. 5, since the authors claim it as one of the best performing PRNGs currently available.

2.2 Transforming Uniformly Distributed PRNs

Most PRNGs only generate uniformly distributed PRNs, but other distributions may be required. A PRNG that supports postprocessing of uniformly distributed PRNs to generate different distributions is an essential feature that should be included: for instance, Gaussian distributed PRNs are often used.

Box-Muller [6] is a common algorithm to generate a pair of independent Gaussian distributed PRNs, based on a set of uniformly distributed numbers.

However, its computational efficiency is flawed due to its iterative nature and reliance on square root, logarithmic, and trigonometric functions.

The inverse transform sampling¹ is a method that transforms uniformly distributed numbers into any distribution, given its cumulative distribution function (CDF). The PRN number can be afterward adjusted to a specific mean and standard deviation, as required by a Gaussian distribution [1, 5, 8, 10]. Current implementations, widely accepted by the scientific community, use an extremely accurate approximation of the Gaussian CDF, which is faster than most transformations. The computational performance of this and Box-Muller methods will be assessed and evaluated on real scientific case studies.

2.3 PRNG Libraries

Most scientific computing libraries and frameworks provide efficient implementations of a wide variety of PRNGs. In the context of the particle physics community, related to the case study presented in Sect. 4, the most popular scientific libraries are provided in the ROOT framework [19]. This framework only offers the Mersenne Twister PRNG with the Box-Muller transformation and is used by default in the three case study variants.

MKL [25] is one of the most popular scientific computing libraries that offers a wide range of mathematical functionalities, also featuring the Mersenne Twister. The Box-Muller and ICDF (inverse transform sampling) transformations are available in this library and will be used to convert uniformly distributed PRNs into a Gaussian distribution. MKL also provides the option to generate a batch of PRNs, whose performance is also evaluated in Sect. 5.

The Standard Template Library (STL) [23] implements a small selection of PRNGs, from which the Mersenne Twister will be considered. However, it is not clear which algorithm STL uses to transform uniform numbers to follow a Gaussian distribution, as chosen by the developers of each C++ compiler.

PCG [18] only supports generation of uniformly distributed PRNs, so it has to be coupled with external implementations of distribution transformations.

PRNG offload to GPU devices is supported through the use of specific libraries, such as cuRAND [17], available in the NVidia CUDA toolkit. It provides an efficient parallel implementation of the Mersenne Twister algorithm and the Box-Muller transformation.

The efficient use of PRNGs on parallel environments, such as multicore or GPU devices, requires multiple independent PRN streams. Several less popular libraries support PRN generation with multiple streams, such as clRNG [13] in OpenCL and RngStreams [12] in C. Adaptations of the latter library to work directly with OpenMP, MPI, and R are also available [9].

¹Explained in https://en.wikipedia.org/wiki/Inverse_transform_sampling.

A new PRN generation in a parallel environment can follow these approaches:

- A single PRNG to feed requests from all concurrent consumer threads, where each PRNG is atomic; results are reproducible as PRNs consumed by each thread vary between runs [7]; no support for PRNG concurrent execution.
- A single PRNG to feed each consumer stream request, using a transition function to guarantee no correlations among streams, known as leapfrog, used in some cuRAND PRNGs [21], supports concurrent execution.
- A single PRNG to feed requests from all concurrent consumer threads, with a different pre-computed seed for each stream, known as block splitting, causing the generated PRNs to be equally spaced in the overall PRN sequence, which may be slow as shown in [3]; it supports concurrent execution.
- An independent PRNG per consumer thread, initialized with different sets of parameters, known as parameterization; if these parameters are not adequate, streams may not be truly independent, as referenced in [4]; it is the most common and portable approach used in scientific code and is implemented by most libraries, such as MKL and cuRAND, and also in libraries specialized for multicore and GPU devices [15, 22].

3 The PRNG-Broker

An implementation of a PRNG on a library is as important to the overall application performance as the approach used to interact with the PRNG itself. For instance, one can generate all PRNs upfront or request a PRN when needed, and these approaches will have a different performance impact depending on the application and execution environment, namely, in parallel code where multiple threads may access shared PRNs and/or PRNGs.

The PRNG-Broker acts as a middle layer between the application code, e.g., a Monte Carlo simulation, and specialized PRNG libraries. It efficiently manages parallel PRN requests to external PRNG libraries, adequately using the computational resources available in multicore, many-core, and GPU devices. This efficient management of PRN generation focuses on improving the performance of parallel compute-bound applications but also provides a significant benefit for both sequential and memory-bound codes.

The key features of the current PRNG-Broker are summarized as follows:

- Select and use a PRNG and associated number distribution per consumer thread, to fill in advance a thread-private dual buffer with a batch of PRNs, for each consumer thread.

- A simple and intuitive API in C++ to hide from the programmer the complexities of the parallel broker.²
- Easy replacement of requests to a PRNG library in the user code by the PRNG-Broker: simply change the instantiation of the PRNG library class.
- Supported generators: statistically tested sequential and parallel implementations of a LCG subset and Mersenne Twister, available in the ROOT, MKL, PCG, STL, and cuRAND libraries; more PRNGs can be easily added.
- Supported distributions of generated PRNs: uniform and Gaussian.
- PRNG parallelization strategies: parameterization and block splitting.
- Hide the overhead of the PRN generation and the memory transfer on PRNG accelerators connected through PCI Express and Ethernet: a thread-private dual-buffer PRN storage to reduce the application waiting time for PRNs.
- Adequate use of the available parallel computing resources in multicore and many-core servers (as well as using additional servers for PRN generation) and many-core and GPU accelerator devices.

3.1 Core of the Broker

The request for PRNs in a multithreaded application can be performed using one of the following approaches:

- To call the PRNG whenever a single PRN is needed.
- To generate a batch of PRNs and store the result in a shared buffer; when a PRN is needed, the consumer thread removes it from the buffer; when the buffer is empty, a new batch is requested.
- Similar to the previous one, but to store the result in a thread-private buffer
- To generate a batch of PRNs and store the result in a thread-private dual-buffer; while the one buffer is being consumed, the other is being filled.

Preliminary results showed that sharing buffers among consumer threads degrades performance, due to thread contention when accessing shared resources. The PRNG with leapfrogging is not supported by the current PRNGs in the PRNG-Broker. Sharing a single sequential PRNG among multiple concurrent threads may affect the statistical quality of the generators; this approach should also be avoided, although it is still a common practice in some simulations.

The key benefit of the dual-buffer PRNG approach is to remove the need to synchronize a contention-free access to the buffer between the broker and a consumer. The PRNG-Broker always stores the generated PRNs in thread-private dual buffers, whose size is user configurable. It offers two alternative approaches to generate batches of PRNs, which can be selected by the user:

²An in-depth description of the PRNG-Broker API and its installation process is available at <https://github.com/prng-broker/prng-broker/wiki/PRNG-Broker>.

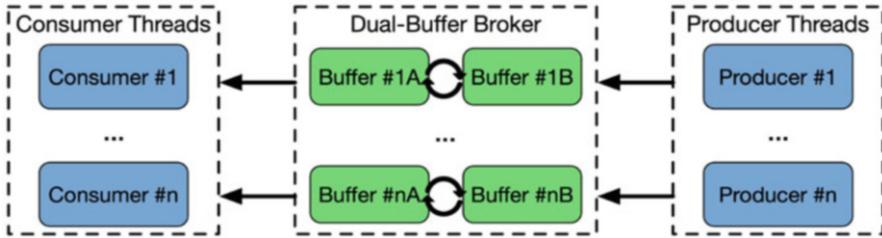


Fig. 1 Dual-buffer broker with multiple PRNG instances using parameterization

- A shared PRNG, where PRNs are placed in the different sets of dual buffers using block splitting
- A PRNG per thread, where each PRNG instance is initialized with different internal-state parameters (not only a different seed; see Fig. 1)

Figure 1 illustrates the dual-buffer approach with multiple PRNG instances, each initialized with a different set of parameters. Each PRNG instance will be executed by a concurrent thread, as preliminary results showed that using a single PRNG degraded performance, and is responsible for the dual buffer of a single consumer thread, which allows simultaneous loads of different consumer thread's buffers. A management thread is responsible for each dual-buffer, swapping buffers once one is consumed and signaling the producer thread to generate a new batch of PRNs for the depleted buffer. The threads on the host application may never wait for PRNs, as the generation latency is hidden by the dual buffer.

3.2 Hardware-Aware PRN Generation

The PRNG-Broker supports *native* and *offload* execution of most PRNGs that it interfaces with, as illustrated in Fig. 2. *Native* execution processes the PRNGs on the computing device running the host application, such as a multicore or many-core device, sharing computational resources with other computational tasks. *Offload* execution processes PRNGs on an alternative computing device, such as a GPU or a many-core accelerator on the same server, or even a separate compute server, freeing computing resources on the host device that can be used to process the user application.

The PRNG-Broker supports PRNG offload to accelerators over PCI Express, using NVidia GPUs or the Intel KNC, or to additional compute servers over a network interconnection. However, using these devices connected introduces the overhead of transferring data from these devices to the host memory space. Offloading PRNGs to other devices/servers may be less efficient, depending on the interconnection, but the dual-buffer approach has the potential to hide the latency of memory transfers.

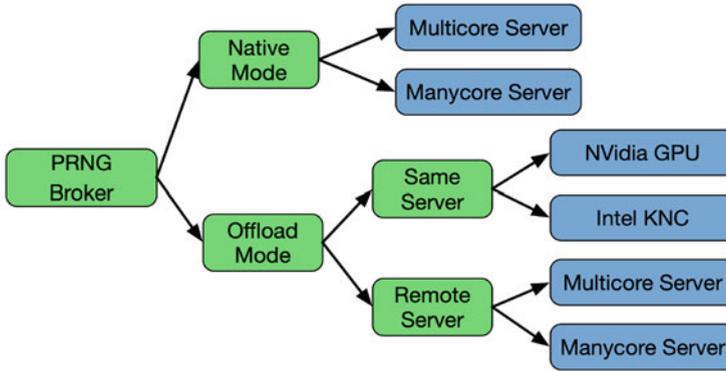


Fig. 2 Hardware configurations supported by the PRNG-Broker library

The management thread allocates the buffers in memory, each with a default capacity for 50K PRNs (configurable by the user), initializes the PRNGs on the devices, and handles data transfers. The overhead of the management threads is minimal as they are only used to swap the buffers and are asleep the remaining of the application runtime. In *native* mode, a thread or a set of threads execute a single PRNG (using block splitting) or multiple PRNG instances (using parameterization) to fill the empty buffers of multiple management threads.

The allocation of the PRN buffers may have a significant impact on performance on NUMA multi-socket servers: when a computing thread has to access a buffer allocated in a memory bank of the companion multicore device, it will suffer a larger latency penalty than accessing a memory bank of the multicore device where the thread is being executed. To mitigate this penalty, management threads are assigned and bound to a core on the same device as the associated computing/consumer thread, ensuring that the buffers are allocated in the memory bank closer to the computing threads.

When offloading PRNGs to accelerator devices, each management thread in the host uses a private stream to launch the PRNGs and to perform the memory transfers, ensuring simultaneous execution. This approach is only used for PRNGs that support parameterization initialization, to guarantee the statistical quality of the generated PRNs.

3.3 Libraries and the PRNG-Broker API

The PRNG-Broker interfaces with a set of libraries to use efficient and statistically tested implementations of PRNGs and distribution transformations for various computing devices. These libraries are linked to PRNG-Broker during installation to use multicore, many-core, and GPU devices. Currently, PRNG-Broker supports

a set of PRNG implementations, based on the Mersenne Twister algorithm, from ROOT, Intel MKL, GNU STL, PCG, and NVidia cuRAND.

The MKL library is used to generate PRNs for Intel multicore servers, Xeon Phi Knights Landing many-core server (native or offloaded over Ethernet), and Knights Corner accelerator (offload over PCI Express), as it provides optimized implementations for these devices. cuRAND is used to generate PRNs for NVidia GPUs (offloaded over PCI Express), as it provides optimized implementations for most generations of GPUs. GNU STL, ROOT, and PCG are used to generate PRNs for multicore and many-core servers (both native and offloaded over Ethernet), but are not the most computationally efficient due to their lack of vectorized and multithreaded code. They are included for compatibility reasons.

The PRNG-Broker API replaces calls to traditional PRNGs to an access to the PRN buffers, hiding the buffer implementation and PRNG parallelization, as shown below:

```
PseudoRandomGenerator rnd;

void some_function () {
    val1 = rnd.gauss(this_thread_id);
    val2 = rnd.uniform(this_thread_id);
}

int main (void) {
    // ... PRNG-Broker initialisation ...
    rnd.init(nthreads); // Amount of computing threads in the user code
    rnd.setGenerator(CURAND); // Any supported PRNG (optional)
    rnd.setSeed(0); // Sets the seed / library pre-computed parameters
    rnd.gaussianConfig(0.0, 0.02); // Avg and stddev if Gaussian is required
    // ... PRNG-Broker cleanup ...
    rnd.shutdown();
}
```

The `gauss` and `uniform` methods return a PRN following either a Gaussian or a uniform distribution, according to the parameters passed to `gaussianConfig`. The `init` method initializes the internal state, such as the number of buffers, the parameters to use by the PRNGs, and the broker itself. The different API options are described at <https://github.com/prng-broker/prng-broker/wiki/PRNG-Broker>.

4 Scientific Data Analyses

A scientific data analysis is a process that converts raw scientific data (often from experimental measurements) into useful information to answer questions, test hypotheses, or prove theories. These applications usually deal with large amounts of experimental data, which is read from one or more files in variable sized chunks or datasets and placed into adequate data structures. Parallel implementations of these analyses, where concurrent threads process independent dataset elements, are often used in scientific data analyses. The use of PRNGs often plays a significant role in performance degradation of these applications.

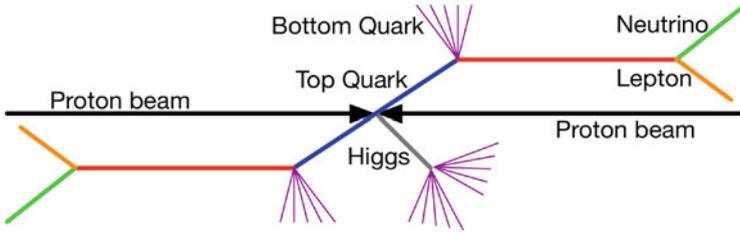


Fig. 3 Schematic representation of the $t\bar{t}$ system and Higgs boson decay

High-energy physics scientists at CERN developed a scientific data analysis code, the $t\bar{t}H$ analysis, to study the associated production of top quarks with the Higgs boson, following head-on proton-proton collisions (known as events) at the Large Hadron Collider (LHC). The final state of an event is recorded by a particle detector, which measures the characteristics of the bottom quarks (detected as jets of particles due to a hadronization process) and leptons (both muons and electrons), but not the neutrinos, as they do not interact with the detector sensors. This final state is presented in Fig. 3.

$t\bar{t}H$ analytically computes the characteristics of the neutrinos with the measured data of other particles, to reconstruct at the end of the data processing both top quarks and the Higgs boson. This process, known as kinematic reconstruction, tests every combination of bottom quarks and leptons, which are stored in a specific structure in predefined files provided by the experiments at the LHC. Three variants of the $t\bar{t}H$ analysis were considered as representative case studies:

- The `ttH_as` (*accurate sensors*) assumes that the data measured by the detector is 100% accurate and requires only 30 PRNs per event.
- The `ttH_sci` (*sensors with a confidence interval*) performs an extensive sampling within the 99% confidence interval in the kinematic reconstruction; this version works with 1024 samples, requiring a total of 30Ki PRNs per event.
- the `ttH_scinp` (*sci with a new pipeline*) performs different operations, maintains the same interstage dependencies, and uses 10Ki PRNs per event.

The PRNG used by default by these data analyses is the Mersenne Twister implementation provided by ROOT using the Box-Muller transformation.

5 Results and Discussion

Once the PRNG-Broker was validated, several aspects were considered for a performance evaluation, such as a characterization of the test beds and associated case studies, and a set of different experimental measurements:

- The throughput of different PRNG algorithms and implementations

- The execution times of the three variants of the case study managed by the PRNG-Broker, varying algorithms, and configurations
- A comparative performance evaluation of the fastest configuration on different server architectures, with and without accelerators

5.1 Test Beds and Associated Case Studies

Three test beds were used for the quantitative evaluation of PRNG-Broker:

- A dual socket server with 12-core Intel Xeon E5-2695v2 Ivy Bridge devices (addressed as IB) at 2.4 GHz, with one NVidia Tesla K20 with 2496 CUDA cores and one Intel Xeon Phi 7120p Knights Corner (addressed as KNC)
- A dual socket server with 16-core Intel Xeon E5-2683v4 Broadwell devices (addressed as BW) at 2.1 GHz;
- A many-core server with 64-core Intel Xeon Phi 7210 Knights Landing device (addressed as KNL) at 1.30 GHz, using the quadrant core clustering and the on-package embedded RAM configured as cache.

A k -best measurement heuristic was used to ensure that the results can be replicated, with $k = 5$ with a 5% tolerance, a minimum/maximum of 15/25 measurements. The multithreaded tests used one computing thread per core, as preliminary tests showed that using core multithreading provided no performance improvements. The test beds used in the multi-server tests are interconnected by a 10 Gbit Ethernet network. The compiler used was Intel Compiler version 18.

The `ttH_sci` application, which is the most PRNG-intensive, spent around 90% of the execution time calling the ROOT PRNG, while the application that required slightly less PRNs (1/3), `ttH_scinp`, spent around 50% of the execution time. `ttH_as` spent less than 10% of its execution time on PRN generation, but it is used to evaluate the overhead of the PRNG-Broker.

5.2 Performance of the Case Studies with the PRNG-Broker

Figure 4 shows the speedup of the three 24-threaded versions of the $t\bar{t}H$ analyses on the dual 12-core server, with the selected PRNG algorithms and different approaches for PRNG execution, vs the original code. The batch MKL generator with the inverse transform sampling method was used as the PRNG representative of the MKL library, due to its performance in preliminary tests.

The PRNGs with the *DB* indication use the dual-buffer approach with the PRNG-Broker while without *DB* indicates that the PRNG is used in single-PRN generation without PRNG-Broker. The efficient use of PRNGs in PRNG-Broker leads to a better performance at several levels:

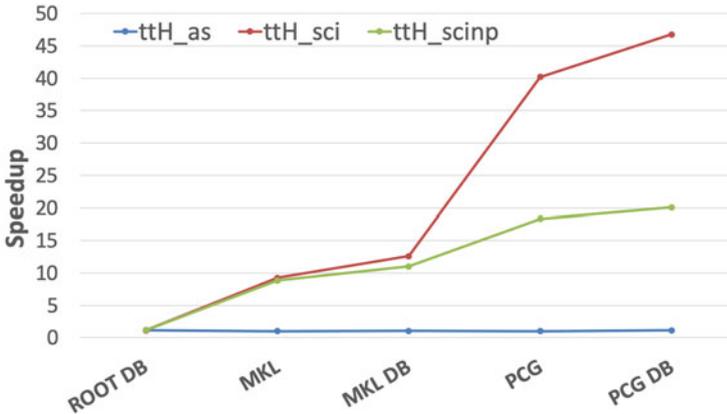


Fig. 4 Speedup of the parallel $t\bar{t}H$ analyses with different PRNG-Broker approaches vs the original ROOT single number PRNG on the 2*12-core IB server

- A significant amount in the high-demanding PRNs `ttH_sci`, when compared to the original code, up to 48x.
- The `ttH_scinp` also benefited from PRNG-Broker, with a speedup improvement up to 20x using PCG.
- The performance of `ttH_as` was not degraded by the use of PRNG-Broker.
- The use of a dual-buffer approach outperformed a single buffer by 15%, 23%, and 18% for the ROOT, MKL, and PCG PRNGs, respectively, and may improve further with parallel code.

However, these improvements are still limited by the hardware resources, since the analysis computing threads are competing for the same computational resources as the PRNG-Broker. Vectorization also had a significant impact on the performance of PRNGs: the generation of a large amount of PRNs adequately explores this feature, which is not present in the single PRN generation.

5.3 Comparative Performance on Different Servers

Offloading PRNG to Kepler or KNC accelerators frees computational resources on the multicore devices to be used by other computations of the $t\bar{t}H$ analyses. Additional KNL and IB servers were also used to offload the PRNG, to ensure that the performance improvements obtained using the accelerators were not only due to the increase in computational resources allocated to the $t\bar{t}H$ analyses. Figure 5 shows the execution times of the parallel case studies on different server configurations, using the dual-buffer PRNG-Broker: (i) on single servers with no accelerators, (ii) on the IB server with an accelerator dedicated to PRNGs, and (iii) on the IB server accessing an external server dedicated to PRNGs.

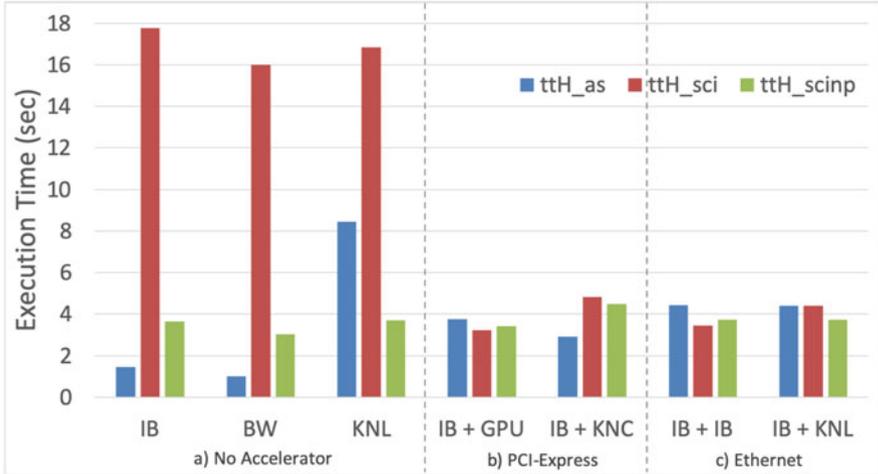


Fig. 5 Execution times of the PRNG-Broker on different server configurations

The performance of the `ttH_sci` benefited from offloading the PRN generation from the host server to an additional server or accelerator devices, with speedups of 5x using the IB server with the Kepler GPU over the BW server. This performance gap would be greater if the Kepler GPU was coupled to a BW server, instead of the IB server, which has eight less computing cores and AVX, instead of AVX2. A similar behavior was observed when offloading the PRN generation to a KNC accelerator or an additional IB and KNL servers. The performance of `ttH_as` and `ttH_scinp` did not improve as much, since the PRN generation accounts for a small portion of their execution time.

The performance of the `ttH_as` application was decreased by 2.9x and 4.4x when using the KNC accelerator or an external KNL server, respectively. Since the code lacks heavy computations and does not need large sets of PRNs, the PRNG-Broker cannot hide the memory transfer costs over PCI Express and Ethernet. The GPU and the external IB server over Ethernet spent >90% of the PRNG time to transfer the PRNs to the host memory, while the KNC accelerator and the external KNL server spent around 80%.

The overall performance of the `ttH_sci` and `ttH_scinp` applications vs the original code that requests successive single PRNs from the inefficient ROOT PRNG, using the Kepler GPU accelerator, was improved: up to 70x and 12x, respectively. An increase of 11x and 12x were due to a faster PRNG and the remainder due to the PRNG-Broker. A similar behavior is observed for the `ttH_sci` and `ttH_scinp` applications with either the KNC accelerator or an external KNL server: speedups of 47x and 51x, respectively. Using for these applications an external IB server to run the PRNGs, the performance improvements were up to 65x and 11x.

These tests proved that applications that require a very large amount of PRNs can greatly benefit by efficiently using PRNGs, regardless of the server architecture and configuration in which they will execute. The overall higher speedups, compared to using only the host server, are also due to the higher availability of the computing cores to perform application-specific computations, since they were freed from generating PRNs. The use of a dual-buffer *vs* a single PRN/buffer contributed to minimize the impact of the PRNs transfer from the accelerators on the same server to the host memory or from external servers: these costs were hidden from the application, since the memory transfers occurred while a buffer still had PRNs in the host memory space.

The PCG PRNG was the best performing when using only multicore devices to process both the application code and PRNG. However, the PCG suite uses a more computationally efficient algorithm than the Mersenne Twister, which may not be a fair comparison. It is the responsibility of the end user to assess if this PRNG should be used over other traditional PRNGs available in PRNG-Broker, which are well accepted and extensively tested by the mathematics' community.

6 Conclusions and Future Work

This paper presented the PRNG-Broker library and associated API that ensures efficient generation and management of large sets of PRNs and their statistical distribution, in a transparent way to the user. It supports PRNG native execution on multicore and many-core devices or offloaded to many-core and GPU accelerator devices or external servers. A detailed analysis of the PRNG-Broker gave an insight into the best way to use PRNGs with real software codes, evaluating the performance of their implementation and the different approaches to use them. PRNG-Broker interfaces with existing libraries to use efficient implementations of popular PRNGs and focus on an adequate management of the generated PRNs to provide significant performance improvements for applications that require large amounts of PRNs.

Three variations of a real scientific data analysis from CERN high-energy physicists were used as case studies: `ttH_sci` and `ttH_scinp`, both compute-bound codes, and `ttH_as`, an I/O-bound code. `ttH_scinp` and `ttH_sci` require 10Ki and 30Ki PRNs per event (dataset element), while `ttH_as` only requires 30, with the tested dataset containing around 800K events. In both single- and dual-buffer approaches, the PRNG is managed by an additional thread per computing thread, so that data processing and the PRNG can be simultaneously executed.

The dual-buffer approach provided speedups over single PRN and single-buffer generation for every PRNG tested in the `ttH_sci` and `ttH_scinp` applications. The most significant improvements were obtained by using external multicore/many-core servers (over Ethernet) and computing accelerators (over PCI Express) to generate the PRNs, as these approaches free the host multicore device

that would be occupied on the PRNGs to process the applications. These approaches provided speedups up to 35x and 71x on single- and dual-socket servers, using the Kepler GPU, as well as up to 51x improvement using an external many-core server for the `ttH_sci` application. The use of better vector operations may have a significant improvement on PRNG performance, as proven by the improvement up to 65% of the `ttH_as` performance on a server with AVX-2 vs a server with AVX instruction sets.

Four main conclusions can be extracted from this analysis:

- The choice of an efficient implementation of a given PRNG is imperative for the application performance: both ROOT and MKL implement the Mersenne Twister, but MKL is, at least, 10x faster.
- The way these PRNGs are used in each application may have a significant impact on performance: the `cuRAND` dual buffer was 2.3x faster than the single-buffer implementation.
- Offloading PRN generation to computing accelerators and other servers using the dual-buffer approach provides a significant performance improvement, as PRNs can be concurrently generated with the application execution.

When using the PRNG-Broker natively on multicore/many-core devices, the execution of the PRNG algorithm shares computational resources with the application, i.e., threads executing the PRNG have to compete with application threads for multicore time. Alternatively, computing cores could be reserved for the PRNG-Broker, but with two caveats: (i) the user would have to ensure that the threads from the application would not use the cores assigned for the PRNG-Broker and (ii) the amount of reserved cores would be dependent on the PRN requirements of each application, which is usually not known without profiling. The former caveat goes against the purpose of PRNG-Broker, which is to hide the complexities of generating and managing PRNs from the user, and would unnecessarily increase the learning curve of the broker. Assuming that this would not be a problem, the latter would require the user to configure the application threads to not use some computing cores, which could vary according to the application and even during the execution, due to dynamic adjustments of the PRNG-Broker. The feasibility of this approach is being evaluated, but preliminary tests already showed no significant improvement for the case studies.

This work focused mainly on the popular Mersenne Twister algorithm, but others could be integrated into PRNG-Broker to extend its range of target applications, such as cryptographically secure PRNGs. The SIMD-oriented Mersenne Twister [20] could also be tested, but it is expected that an efficient SIMD implementation for Intel multicore devices is provided by MKL, as proved by the initial benchmark results.

References

1. S. Asmussen, P.W. Glynn, *Stochastic Simulation: Algorithms and Analysis*, vol. 57 (Springer Science & Business Media, 2007)
2. ATLAS Collaboration: Observation of a new particle in the search for the Standard Model Higgs Boson with the ATLAS detector at the LHC. *Phys. Lett. B* **716**(1), 1–29 (2012)
3. T. Bradley, J. du Toit, R. Tong, M. Giles, P. Woodhams, Parallelization techniques for random number generators, in *GPU Computing Gems Emerald Edition* (Elsevier, 2011), pp. 231–246
4. K. Claessen, M.H. Pałka, Splittable pseudorandom number generators using cryptographic hashing, in *ACM SIGPLAN Notices*, vol. 48 (ACM, 2013), pp. 47–58
5. L. Devroye, Sample-based Non-uniform random variate generation, in *Proceedings of the 18th Conference on Winter Simulation* (ACM, 1986), pp. 260–265
6. E. Golder, J. Settle, The Box-Muller method for generating pseudo-random normal deviates. *Appl. Stat.* **25**, 12–20 (1976)
7. D.R. Hill, C. Mazel, J. Passerat-Palmbach, M.K. Traore, Distribution of random streams for simulation practitioners. *Concurr. Comput. Pract. Exp.* **25**(10), 1427–1442 (2013)
8. W. Hörmann, J. Leydold, G. Derflinger, *Automatic Nonuniform Random Variate Generation* (Springer Science & Business Media, 2013)
9. A.T. Karl, R. Eubank, J. Milovanovic, M. Reiser, D. Young, Using RngStreams for parallel random number generation in C++ and R. *Computat. Stat.* **29**(5), 1301–1320 (2014)
10. A.M. Law, W.D. Kelton, W.D. Kelton, *Simulation Modeling and Analysis*, vol. 3 (McGraw-Hill, New York, 2000)
11. P. L’Ecuyer, R. Simard, TestU01: AC library for empirical testing of random number generators. *ACM Trans. Math. Software* **33**(4), 22:1–22:40 (2007)
12. P. L’Ecuyer, R. Simard, RngStreams: An object-oriented random-number package in c with many long streams and substreams. *Oper. Res.* **50**(6), 1073–1075 (2012)
13. P. L’Ecuyer, D. Munger, N. Kemerchou, clRNG: A random number API with multiple streams for OpenCL. Technical report, University of Montreal (2015)
14. G. Marsaglia, The Marsaglia Random Number CDROM Including the Diehard Battery of Tests of Randomness. Computer file, Florida State University (1995)
15. M. Mascagni, A. Srinivasan, Algorithm 806: SPRNG: A scalable library for pseudorandom number generation. *ACM Trans. Math. Softw.* **26**(3), 436–461 (2000)
16. M. Matsumoto, T. Nishimura, Mersenne Twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.* **8**(1), 3–30 (1998)
17. Nvidia: CURAND library. NVIDIA Corporation (2010)
18. M.E. O’Neill, PCG: A Family of Simple Fast Space-Efficient Statistically Good Algorithms for Random Number Generation. Technical Report, HMC-CS-2014-0905, Harvey Mudd College, Claremont (2014)
19. F. Rademakers, ROOT — A C++ framework for petabyte data storage, statistical analysis and visualization. *Comput. Phys. Commun.* **180**(12), 2499–2512 (2009)
20. M. Saito, M. Matsumoto, SIMD-oriented fast mersenne twister: A 128-bit pseudorandom number generator, in *Monte Carlo and Quasi-Monte Carlo Methods 2006* (Springer, 2008), pp. 607–622
21. M. Saito, M. Matsumoto, A deviation of CURAND: Standard pseudorandom number generator in CUDA for GPGPU, in *Proceedings of 10th International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing* (2012), pp. 92–100
22. M. Saito, M. Matsumoto, Variants of Mersenne twister suitable for graphic processors. *ACM Trans. Math. Softw.* **39**(2), 12:1–12:20 (2013)
23. for Standardization, I.O., the International Electrotechnical Commission: ISO/IEC 9899:2018. Tech. rep., International Organization for Standardization and the International Electrotechnical Commission, Geneva (2018)

24. J. Von Neumann, Various techniques used in connection with random digits. *Natl. Bur. Stand. Appl. Math. Ser.* **12**, 36–38 (1951)
25. E. Wang, Q. Zhang, B. Shen, G. Zhang, X. Lu, Q. Wu, Y. Wang, *Intel Math Kernel Library* (Springer, 2014)

Numerical Modeling of a Viscous Incompressible Fluid Flow in a Channel with a Step



Saeed M. Dubas, Paul Bouthellier, Nihal Siriwardana, and Laura Wieserman

1 Introduction

The problem of steady incompressible flow of gases and liquids in a channel with a step has been the subject of many studies in computational fluid dynamics. Panovko [5] used the method of separation of physical factors to model three-dimensional flow of a viscous fluid in a channel with a step at the Reynolds number $R = 100$. He reported circulatory flow near the step, which decreased in intensity toward the channel outlet. Two benchmark problems that are documented in international workshop proceedings [7, 8] are steady expansion flows and contraction flows in channels with a step. Boger [2] provided a comprehensive review for contraction flows for channels with steps. For the flow into a symmetrical contraction in the form of a step, his experimental data suggest the formation of a “trailing edge” vortex downstream of the step, and detecting this vortex posed a major test for a numerical scheme. Dennis and Smith [9] used a method based on central differences, but were unable to detect the trailing edge vortex visually downstream of the step. However, through grid refinement, they were able to infer qualitatively the presence of this vortex. Their results have been supported experimentally by Durst and Loy [10]. Hawken et al. [6] used a Taylor–Galerkin algorithm and were able to visually detect the trailing edge vortex at the Reynolds number $R = 450$. They went up to Reynolds

S. M. Dubas (✉) · L. Wieserman
Engineering and Computer Science Division, University of Pittsburgh at Johnstown, Johnstown, PA, USA
e-mail: dubasis@pitt.edu

P. Bouthellier
Department of Mathematics, University of Pittsburgh at Greensburg, Greensburg, PA, USA

N. Siriwardana
Department of Mathematics, Prairie View A & M University, Prairie View, TX, USA

number $R = 100$ for the expansion case and $R = 800$ for the contraction case. We obtained converged results up to Reynolds number $R = 1000$.

This chapter uses a stable fourth-order central difference method [4] by writing Taylor's series expansions of the error terms in a form that produces strong central coefficients.

Our work supports experimental evidence, suggested by Boger [2], that a trailing edge vortex is formed downstream of the step, which can be seen in our plotted stream functions at $R = 200$ to $R = 400$, and later at $R = 1000$. Onur and Baydar [1] carried out experimental work on this problem, and showed photographs for $R = 200$, which are in agreement with our results. Our results are also in good agreement with the tabulated results of Greenspan [3], requiring a relatively coarse grid for the same accuracy.

2 Mathematical Problem

Consider the flow of a viscous incompressible fluid in a channel with a step. The flow is governed by the steady-state Navier–Stokes equations defined by

$$\Delta \psi = -\omega \quad (1)$$

$$\Delta \omega + R (\psi_x \omega_y - \psi_y \omega_x) = 0 \quad (2)$$

These equations are valid in the interior of the domain in Fig. 1, where ψ and ω are the stream and the vorticity functions respectively, and R is a flow parameter called the Reynolds number. The boundary conditions to be satisfied are

$$\psi = 1, \quad \psi_y = 0, \quad \text{on } HG \quad (3)$$

$$\psi = 0, \quad \psi_y = 0, \quad \text{on } AB, CD, EF \quad (4)$$

$$\psi = 0, \quad \psi_x = 0, \quad \text{on } BC, DE \quad (5)$$

$$\psi = 3y^2 - 2y^3, \quad \omega = 12y - 6, \quad \text{on } AH \quad (6)$$

$$\psi_x = 0, \quad \omega_x + R\psi_y (\omega + \psi_{yy}) = 0 \quad \text{on } FG \quad (7)$$

Conditions (6) are those of Poiseuille flow, while conditions (7) make the flow horizontal and the pressure constant on FG.

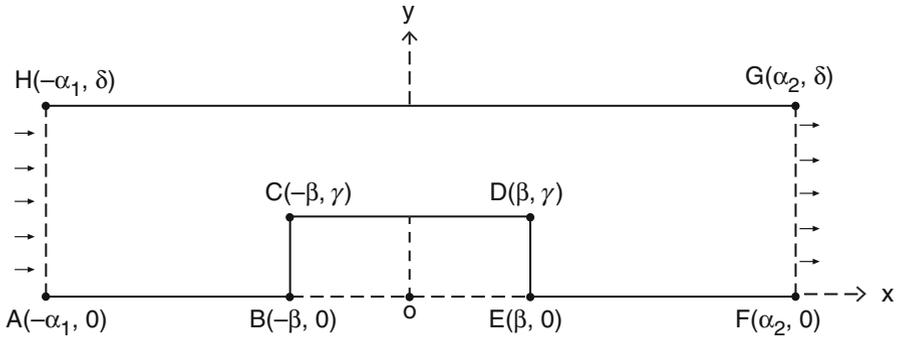
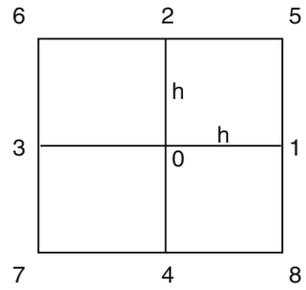


Fig. 1 The channel with a step

Fig. 2 The grid



3 The Difference Schemes

We present the difference methods for the given differential equations (1)–(2) as a special case of the second-order elliptic equation.

$$Lu \equiv u_{xx} + u_{yy} + p(x, y) \psi_x + q(x, y) u_y + r(x, y) u = s(x, y) \tag{8}$$

Note that since $r(x, y) = 0$ for both (1) and (2), we may write (8) as

$$Lu \equiv u_{xx} + u_{yy} + pu_x + qu_y = s \tag{9}$$

where $p = p(x, y)$, $q = q(x, y)$, and $s = s(x, y)$.

Also, note that

$$p(x, y) = q(x, y) = 0, \quad s(x, y) = -\omega \text{ for (1),}$$

$$p(x, y) = -R\psi_y, \quad q(x, y) = R\psi_x, \quad s(x, y) = 0 \text{ for (2).}$$

Using the grid shown in Fig. 2, which shows the placement of nine points, we obtain the following approximations at the point (x_i, y_j) , numbered 0 in Fig. 2.

$$u_{xx} = \frac{u_3 - 2u_0 + u_1}{h^2} - \frac{h^2}{12}u_{xxxx} + O(h^4) \quad (10)$$

$$u_{yy} = \frac{u_4 - 2u_0 + u_{21}}{h^2} - \frac{h^2}{12}u_{yyyy} + O(h^4) \quad (11)$$

$$u_x = \frac{u_1 - u_3}{2h} - \frac{h^2}{6}u_{xxx} + O(h^4) \quad (12)$$

$$u_y = \frac{u_4 - u_2}{2h} - \frac{h^2}{6}u_{xyy} + O(h^4) \quad (13)$$

Substituting (10)–(13) into the differential Eq. (9), we obtain the difference operator L_h , defined by

$$L_h u_0 \equiv \sum_{i=0}^4 \alpha_i u_i = s_0 + E_0(u) \quad (14)$$

where $E_0(u)$ is the truncation error given by

$$E_0(u) = \frac{h^2}{12} (u_{xxxx} + 2pu_{xxx} + u_{yyyy} + 2pu_{yyy}) + O(h^4) \quad (15)$$

and

$$\begin{aligned} \alpha_0 &= -\frac{4}{h^2} \\ \alpha_1 &= \frac{1}{h^2} + \frac{p_0}{2h} \\ \alpha_2 &= \frac{1}{h^2} + \frac{q_0}{2h} \\ \alpha_3 &= \frac{1}{h^2} - \frac{p_0}{2h} \\ \alpha_4 &= \frac{1}{h^2} - \frac{q_0}{2h} \end{aligned} \quad (16)$$

Note that when $p(x, y) = -R\psi_y$ or $q(x, y) = R\psi_x$ is large, that is, when R is large, the central difference coefficient α_0 is small relative to the other coefficients, which is the main cause of instability of the central difference method.

To obtain a stable fourth-order operator, we rewrite the central difference approximation (13) in a form that includes the error terms. Therefore,

$$L_h u_0 - E_0(u) = s_0 \quad (17)$$

We next denote the error term $E_0(u)$ in terms of the lower order derivatives given by (10)–(13), and mixed partial derivatives u_{xy} , etc., which can be denoted by the nine points grid in Fig. 2, with a stabilizing effect. From (9), we have

$$u_{xx} = -(pu_x + qu_y - u_{yy}) + s \quad (18)$$

Differentiating both sides of this equation with respect to x to obtain u_{xxx} and u_{xxxx} , we have

$$\begin{aligned} u_{xxxx} + 2pu_{xxx} = & - \left[(p^2 + 2p_x)u_{xx} + (pp_x + p_{xx})u_x \right. \\ & + (pq_x + q_{xx})u_y + (pq + 2q_x)u_{yx} + pu_{yyx} + qu_{yxx} \\ & \left. + u_{yyxx} \right] + ps_x + s_{xx} \end{aligned} \quad (19)$$

Similarly, we have

$$\begin{aligned} u_{yyyy} + 2qu_{yyy} = & - \left[(q^2 + 2q_y)u_{yy} + (qq_y + q_{yy})u_y \right. \\ & + (qp_y + p_{yy})u_x + (pq + 2p_y)u_{xy} + qu_{xxy} + pu_{xyy} \\ & \left. + u_{xxyy} \right] + qs_y + s_{yy} \end{aligned} \quad (20)$$

Substituting (19) and (20) into (15), we have

$$\begin{aligned} E_0(u) = & -\frac{h^2}{12} \left[(p^2 + 2p_x)u_{xx} + (pp_x + qp_y + p_{xx} + p_{yy})u_x \right. \\ & + (q^2 + 2q_y)u_{yy} + (pq_x + qq_y + q_{xx} + q_{yy})u_y \\ & + 2(pq + p_y + q_x)u_{xy} + 2(pu_{xyy} + qu_{xxy} + u_{xxyy}) \\ & \left. + \frac{h^2}{12} (ps_x + qs_y + s_{xx} + s_{yy})u_y + O(h^4) \right] \end{aligned} \quad (21)$$

Note that the terms $(p^2 + 2p_x)u_{xx}$ and $(q^2 + 2q_y)u_{yy}$ can produce strong central coefficients when approximated by (10) and (11).

Substituting (21) into (17), we obtain

$$L_h u_0 - \bar{E}_0(u) = s_0^* \quad (22)$$

where

$$\bar{E}_0(u) = E_0(u) - \frac{h^2}{12} (ps_x + qs_y + s_{xx} + s_{yy}) \quad (23)$$

and

$$s_0^* = s_0 + \frac{h^2}{12} (ps_x + qs_y + s_{xx} + s_{yy}) \quad (24)$$

We approximate $\bar{E}_0(u)$ using (10)–(13), whereas the mixed partial derivatives u_{xy} , u_{xyx} , u_{yyx} , u_{xyy} can be readily approximated using Taylor series expansion at the nine point grid, which leads to

$$\bar{E}_0(u) = \sum_{i=0}^8 \beta_i u_i + O(h^4) \quad (25)$$

where

$$\begin{aligned} \beta_0 &= -\frac{4}{6h^2} + \frac{1}{6} (p^2 + q^2 + 2p_x + 2q_y) \\ \beta_1 &= \frac{2}{6h^2} + \frac{p_0}{6h} - \frac{1}{12} (p^2 + 2p_x) - \frac{h}{24} (pp_x + qp_y + p_{xx} + p_{yy}) \\ \beta_2 &= \frac{2}{6h^2} + \frac{q_0}{6h} - \frac{1}{12} (q^2 + 2q_y) - \frac{h}{24} (pq_x + qq_y + q_{xx} + q_{yy}) \\ \beta_3 &= \frac{2}{6h^2} - \frac{p_0}{6h} - \frac{1}{12} (p^2 + 2p_x) + \frac{h}{24} (pp_x + qp_y + p_{xx} + p_{yy}) \\ \beta_4 &= \frac{2}{6h^2} - \frac{q_0}{6h} - \frac{1}{12} (q^2 + 2q_y) + \frac{h}{24} (pq_x + qq_y + q_{xx} + q_{yy}) \\ \beta_5 &= \frac{-1}{6h^2} - \frac{p_0 + q_0}{12h} - \frac{pq + p_y + q_x}{24} \\ \beta_6 &= \frac{-1}{6h^2} + \frac{q_0 - p_0}{12h} + \frac{pq + p_y + q_x}{24} \\ \beta_7 &= \frac{-1}{6h^2} - \frac{p_0 - q_0}{12h} - \frac{pq + p_y + q_x}{24} \\ \beta_8 &= \frac{-1}{6h^2} + \frac{q_0 - p_0}{12h} + \frac{pq + p_y + q_x}{24} \end{aligned} \quad (26)$$

Substituting (25) into (22), we obtain a stable fourth-order operator L_h^* for (9), defined by

$$L_h^* u_0 \equiv \sum_{i=0}^4 u_i^* = s_0^* + E_0^*(u) \quad (27)$$

where

$$\begin{aligned} E_0^*(u) &= h^4 \left[\frac{p^2 u_{xxxx}(\zeta_1, \eta_1) + q^2 u_{yyyy}(\zeta_2, \eta_2)}{144} \right. \\ &\quad \left. + 2pqh^4 (u_{xxyy}(\zeta_3, \eta_3) + u_{xyyy}(\zeta_4, \eta_4)) \right] \end{aligned} \quad (28)$$

and s_0^* is given by (24), and $\alpha_i^* = \alpha_i - \beta_i$.

We also see that the local truncation error $E_0^*(u)$ is $O(h^4)$.

We now set up the difference equations for (1) and (2). For (1), $p = q = 0$ and $s = -\omega$, which leads to the following difference equation

$$L_h^* \psi_0 \equiv \sum_{i=0}^4 \alpha_i^* \psi_i = s_0^* + E_0^*(u) \quad (29)$$

where

$$\begin{aligned} \alpha_0^* &= -\frac{20}{6h^2} \\ \alpha_1^* &= \alpha_2^* = \alpha_3^* = \alpha_4^* = \frac{4}{6h^2} \\ \alpha_5^* &= \alpha_6^* = \alpha_7^* = \alpha_8^* = \frac{1}{6h^2} \end{aligned} \quad (30)$$

and

$$s_0^* = -\omega_0 - \frac{h^2}{12} (\omega_{xx} + \omega_{yy}) \quad (31)$$

Next, for (2), $p = -R\psi_y$, $q = R\psi_x$, and $s = 0$, which results in the following difference equation

$$L_h^* \omega_0 \equiv \sum_{i=0}^4 \alpha_i^* \omega_i = s_0^* + E_0^*(u) \quad (32)$$

where

$$\begin{aligned} \alpha_0 &= -\frac{20}{6h^2} - \frac{1}{6} (p^2 + q^2 + 2p_x + 2q_y) \\ \alpha_1 &= \frac{4}{6h^2} + \frac{p_0}{3h} + \frac{1}{12} (p^2 + 2p_x) + \frac{h}{24} (pp_x + qp_y + p_{xx} + p_{yy}) \\ \alpha_2 &= \frac{4}{6h^2} + \frac{q_0}{3h} + \frac{1}{12} (q^2 + 2q_y) + \frac{h}{24} (pq_x + qq_y + q_{xx} + q_{yy}) \\ \alpha_3 &= \frac{4}{6h^2} - \frac{p_0}{3h} + \frac{1}{12} (p^2 + 2p_x) - \frac{h}{24} (pp_x + qp_y + p_{xx} + p_{yy}) \\ \alpha_4 &= \frac{4}{6h^2} - \frac{q_0}{3h} + \frac{1}{12} (q^2 + 2q_y) - \frac{h}{24} (pq_x + qq_y + q_{xx} + q_{yy}) \\ \alpha_5 &= \frac{1}{6h^2} + \frac{p_0 + q_0}{12h} + \frac{pq + p_y + q_x}{24} \\ \alpha_6 &= \frac{1}{6h^2} + \frac{q_0 - p_0}{12h} - \frac{pq + p_y + q_x}{24} \\ \alpha_7 &= \frac{1}{6h^2} - \frac{p_0 + q_0}{12h} + \frac{pq + p_y + q_x}{24} \\ \alpha_8 &= \frac{1}{6h^2} - \frac{q_0 - p_0}{12h} - \frac{pq + p_y + q_x}{24} \end{aligned} \quad (33)$$

4 Comparison of Results

We ran the present scheme and the scheme of Greenspan [3] at each of the indicated points $(x, y) = (0.1, 0.1)$, and $(x, y) = (1.9, 0.8)$ for ψ and ω , $\alpha_1 = \alpha_2 = 4$, $\gamma = 0.5$, $\delta = 1$, various values of the step size h and Reynolds number R . To check the efficiency, we compared the results at the same points for both methods, as given by Tables 1A, 1B, 2A, 2B, 3A, and 3B.

Table 1A Stream values at $R = 50$

Present (0.1,0.1)		Greenspan (0.1,0.1)		Present (1.9,0.8)		Greenspan (1.9,0.8)	
R	Ψ	Ψ	Ψ	Ψ	Ψ	Ψ	Ψ
$h = 0.1$	2.780 E-02	2.841 E-02	0.8812	0.8775			
$h = 0.05$	2.797 E-02	2.813 E-02	0.8825	0.8815			
$h = 0.025$	2.797 E-02	2.802 E-02	0.8829	0.8824			
$h = 0.0125$	2.797 E-02	2.798 E-02	0.8831	0.8829			
$h = 0.00625$	2.797 E-02	2.798 E-02	0.8831	0.8830			

Table 1B Vorticity values at $R = 50$

Present (0.1,0.1)		Greenspan (0.1,0.1)		Present(1.9,0.8)		Greenspan (1.9,0.8)	
R	ω	ω	ω	ω	ω	ω	ω
$h = 0.1$	-48E-01	-4732 E-01	3567 E-01	3529 E-01			
$h = 0.05$	-4798 E-01	-4776 E-01	3609 E-01	3593 E-01			
$h = 0.025$	-4798 E-01	-4792 E-01	3610 E-01	3608 E-01			
$h = 0.0125$	-4799 E-01	-4797 E-01	3611 E-01	3612 E-01			
$h = 0.00625$	-4799 E-01	-4799 E-01	3612 E-01	3611 E-01			

Table 2A Stream values at $R = 100$

Present (0.1,0.1)		Greenspan (0.1,0.1)		Present(1.9,0.8)		Greenspan (1.9,0.8)	
R	Ψ	Ψ	Ψ	Ψ	Ψ	Ψ	Ψ
$h = 0.1$	2.8 E-02	2.822 E-02	0.8765	0.8801			
$h = 0.05$	2.794 E-02	2.805 E-02	0.8772	0.8757			
$h = 0.025$	2.794 E-02	2.797 E-02	0.8777	0.8772			
$h = 0.0125$	2.794 E-02	2.795 E-02	0.8779	0.8776			
$h = 0.00625$	2.794 E-02	2.794 E-02	0.8780	0.8778			

Table 2B Vorticity values at $R = 100$

Present (0.1,0.1)		Greenspan (0.1,0.1)		Present(1.9,0.8)		Greenspan (1.9,0.8)	
R	ω	ω	ω	ω	ω	ω	ω
$h = 0.1$	-48E-01	-4749 E-01	3495 E-01	3417 E-01			
$h = 0.05$	-4797 E-01	-4782 E-01	3510 E-01	3494 E-01			
$h = 0.025$	-4799 E-01	-4794 E-01	3516 E-01	3512 E-01			
$h = 0.0125$	-4799 E-01	-4798 E-01	3517 E-01	3517 E-01			
$h = 0.00625$	-4799 E-01	-4799 E-01	3519 E-01	3518 E-01			

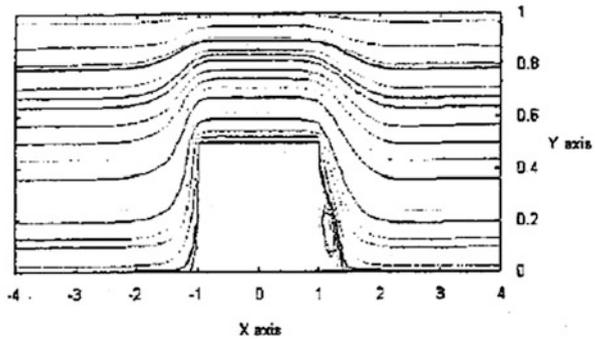
Table 3A Stream values at $R = 1000$

Present (0.1,0.1)		Greenspan (0.1,0.1)		Present(1.9,0.8)		Greenspan (1.9,0.8)	
R	Ψ	Ψ	Ψ	Ψ	Ψ	Ψ	Ψ
$h = 0.1$	2.730 E-02	2.727 E-02	0.8594	0.8412			
$h = 0.05$	2.768 E-02	2.750 E-02	0.8477	0.8403			
$h = 0.025$	2.762 E-02	2.758 E-02	0.8539	0.8539			
$h = 0.0125$	2.760 E-02	2.760 E-02	0.8542	0.8549			
$h = 0.00625$	2.760 E-02	2.760 E-02	0.8545	0.8549			

Table 3B Vorticity values at $R = 1000$

Present (0.1,0.1)		Greenspan (0.1,0.1)		Present(1.9,0.8)		Greenspan (1.9,0.8)	
R	ω	ω	ω	ω	ω	ω	ω
$h = 0.1$	-4716 E-01	-4799 E-01	5733 E-01	2956 E-01			
$h = 0.05$	-4793 E-01	-4808 E-01	3105E-01	3064 E-01			
$h = 0.025$	-4806 E-01	-4809 E-01	3114 E-01	3109 E-01			
$h = 0.0125$	-4808 E-01	-4809E-01	3116E-01	3122E-01			
$h = 0.00625$	-4808 E-01	-4809 E-01	3120 E-01	3124 E-01			

Fig. 3 Plot of ψ for $h = 0.1$, $R = 10$



Results of both methods converged, but for the stream function at $R = 50$ and $R = 1000$, we obtained four digit accuracy for $h = 0.05$, whereas Greenspan scheme required $h = 0.00625$ for the same accuracy.

At $R = 1000$, we obtained four digit accuracy at step size $h = 0.05$, whereas the Greenspan’s scheme required $h = 0.00125$ for the same accuracy.

Using the data obtained from the present scheme, the stream curves are plotted in Figs. 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, and 14, whereas vorticity curves are plotted in Figs. 15, 16, 17, and 18.

From the graphs, we can see that the role of the step size is limited for lower values of R , like $R = 10$ and $R = 50$ in Figs. 3 and 4, but when the value of R becomes 100 or 1000, the step size plays an important role, which is obvious from Figs. 9 and 10, where there is a counterflow on the left-hand side of the channel when step size $h = 0.025$. However, for larger values of h , there is no counterflow on the left-hand side of the channel for the same value of $R = 100$ (see Figs. 7 and 8).

Fig. 4 Plot of ψ for $h = 0.1$, $R = 50$

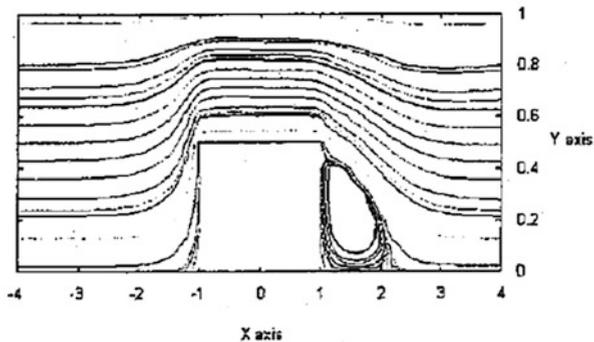


Fig. 5 Plot of ψ for $h = 0.05$, $R = 50$

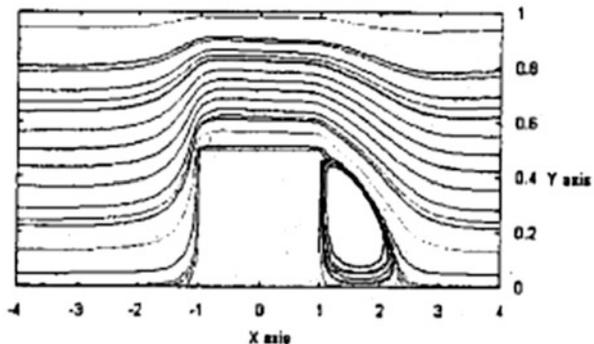
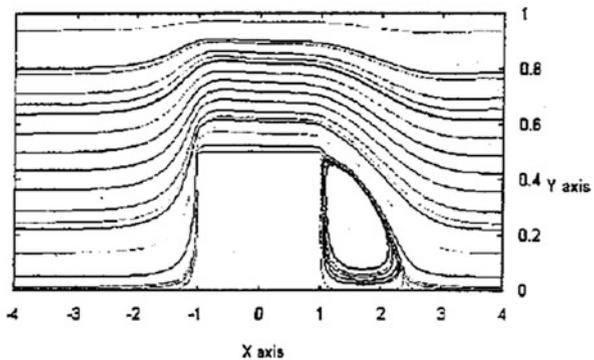


Fig. 6 Plot of ψ for $h = 0.025$, $R = 50$



In the present scheme, counterflow to the left-hand side of the channel appears for the first time when $R = 100$ and the step size is as small as $h = 0.025$, as shown in Fig. 9. Onur and Baydar [1] did experimental work on this problem and showed photographs for $R = 200$. Our results are in agreement with those photographs. Experimental evidence presented by Boger [2] suggests that a trailing edge vortex is formed downstream of the step, which can be seen in our plotted stream functions for $R = 200, 300, 400$, and $h = 0.0125$ in Figs. 13, 14, and 15. It can also be seen in the $R = 1000$ case, see Fig. 10.

Fig. 7 Plot of ψ for $h = 0.1$, $R = 100$

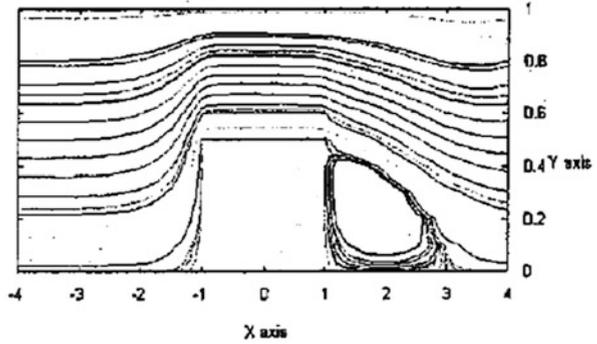


Fig. 8 Plot of ψ for $h = 0.05$, $R = 100$

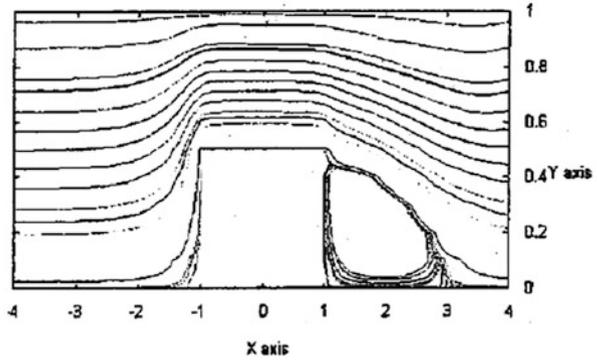
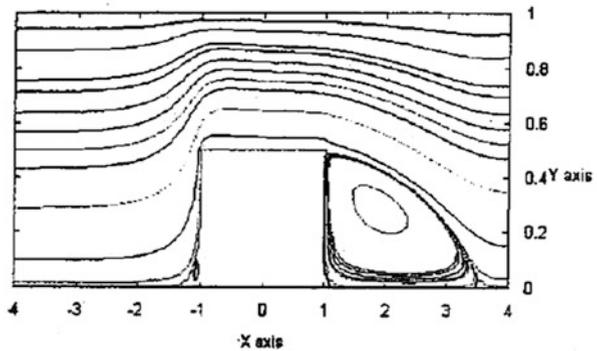


Fig. 9 Plot of ψ for $h = 0.025$, $R = 100$



4.1 Effect of Step Height on the Flow

We checked the effect of varying the step height on the flow, where D denotes the height of the channel at the step ($D < 1$, maximum height).

When $D = 0.25$, it can be seen from the graph in Fig. 11 that flow is everywhere in the channel. There is a large counterflow on the right-hand side and a small counter flow on the left-hand side of the step as compared to the standard case when

Fig. 10 Plot of ψ for $h = 0.025, R = 1000$

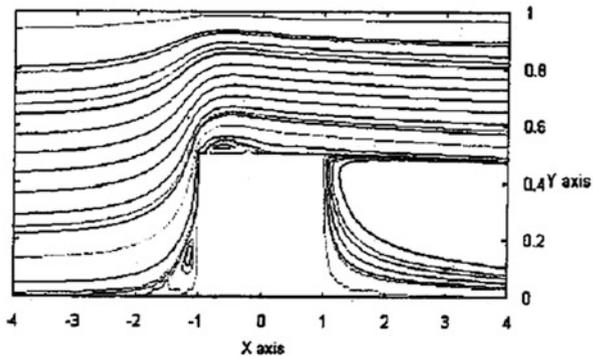


Fig. 11 Plot of ψ for $h = 0.05, D = 0.25, R = 100$

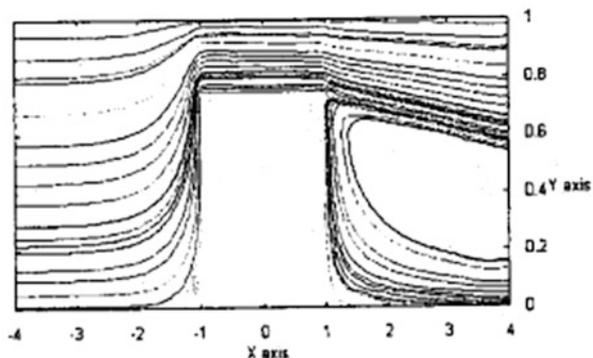
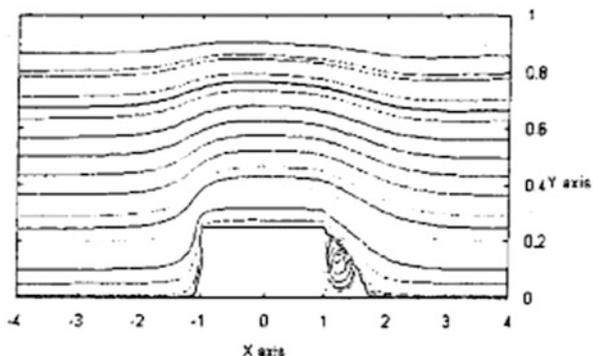


Fig. 12 Plot of ψ for $h = 0.05, D = 0.75, R = 100$



$D = 0.5$, where there is a small counter flow on the right-hand side and no counter flow on the left-hand side of the step, as seen from Fig. 8.

On increasing the value of D to 0.75, as in Fig. 12, flow can be seen everywhere in the channel. There is a small counterflow on the right-hand side of the step as compared to the standard case when $D = 0.5$. Also, refer to Fig. 19 for other interesting results.

Fig. 13 Plot of ψ for $h = 0.0125, R = 200$

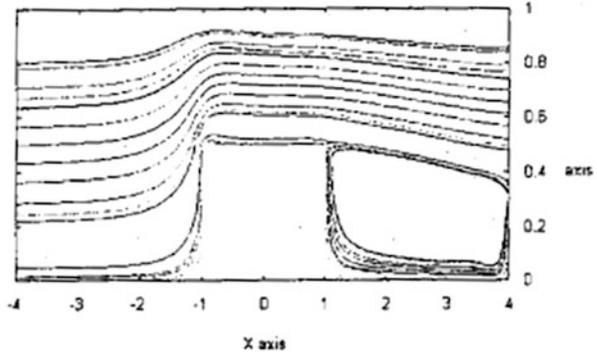


Fig. 14 Plot of ψ for $h = 0.0125, R = 300$

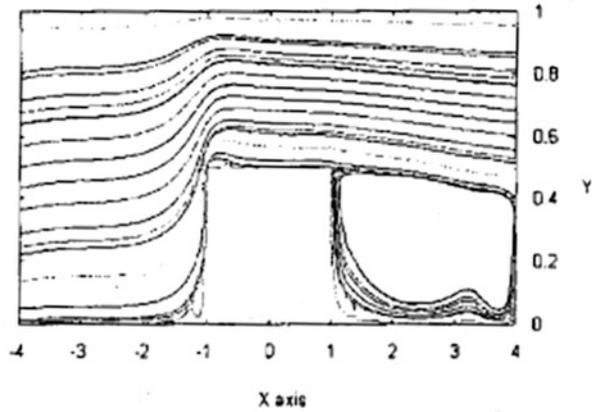
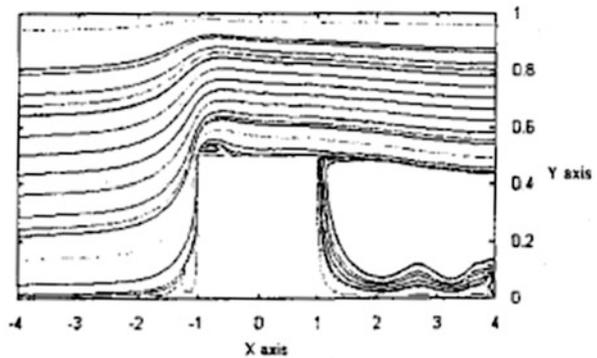


Fig. 15 Plot of ψ for $h = 0.0125, R = 400$



5 Conclusion

In this chapter, we used a high-order numerical scheme based on Choo and Schultz's work [4] to study the flow of a viscous incompressible fluid in a channel with a step. The numerical scheme expressed the error terms of the central difference method in

Fig. 16 Plot of ω for $h = 0.05, R = 50$

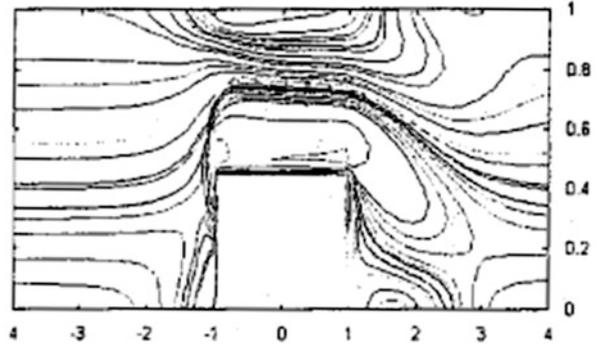


Fig. 17 Plot of ω for $h = 0.05, R = 100$

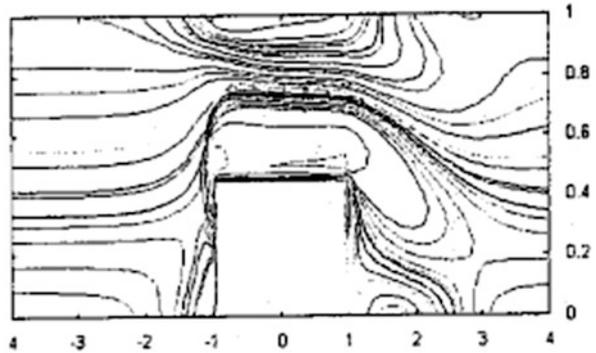
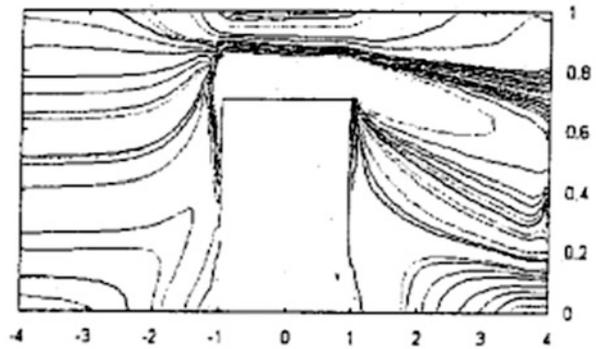
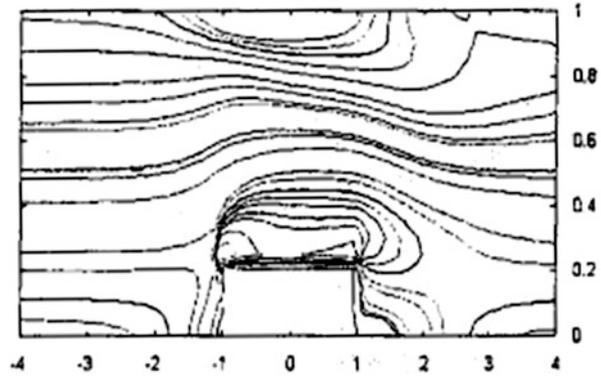


Fig. 18 Plot of ω for $h = 0.05, D = 0.25, R = 1000$



such a form that led to a stable fourth-order operator. The results obtained were in good agreement with the experimental, graphical, and tabulated results obtained in other studies.

Fig. 19 Plot of ω for $h = 0.05, D = 0.75, R = 100$



References

1. H.S. Onur, Baydar, Laminar channels flow over a square step. *Int. J. Eng. Sci.* **30**(9), 1109–1116 (1992)
2. D.V. Boger, Viscoelastic flow through contractions. *Ann. Rev. Fluid. Mech.*, 157–182 (1987)
3. D. Greenspan, Numerical studies of steady, viscous, incompressible flow in a channel with a step. *J. Eng. Math.* **3**, 21–28 (1969)
4. J.Y. Choo, D. Schultz, A high order difference method for the steady state Navier –Stokes equations. *Comput. Math. Applic.* **27**(11), 105–119 (1994)
5. M.Y. Panovko, Numerical modelling of three dimensional flows of a viscous incompressible fluid in a channel with a step. translated from *Teplofizika Vysokikh Temperetur* **27**(6), 1126–1131 (1989)
6. D.M. Hawken, P. Townsend, M.F. Webster, Numerical simulations of viscous flows in channels with a step. *J. Comput. Fluids* **20**(1), 59–75 (1991)
7. *Proc. 6th Mtg of the IAHR Working Group on Refined Modelling of Flows* (Karlsruhe, 1983)
8. K. Morgan, J. Periaux, F. Thomasset (eds.), *Analysis of Laminar Flow Over a Backward Facing Step* (Proc. GAMM Wkshp, Bievres, 1983)
9. S.C.R. Dennis, F.T. Smith, Steady flow through a channel with a symmetrical constriction in the form of a step. *Proc. Royal Soc. Lond.* **A327**, 393 (1980)
10. F. Durst, T. Loy, Investigations of Laminar flow in a pipe with sudden constriction of cross sectional area. *Computers Fluids* **13**(1), 15 (1985)

Modeling, Simulation, and Verification for Structural Stability and Vibration Reduction of Gantry Robots for Shipyard Welding Automation Using ANSYS Workbench[®] and Recurdyn[®]



Seung Min Bae, Won Jee Chung, Hui Geon Hwang, and Yeon Joo Ahn

1 Introduction

With the strengthening of domestic and foreign environmental regulations in recent times, the disposal of small scrapped FRP (fiber-reinforced plastic) vessels is raising a problem and there is a growing interest in eco-friendly aluminum vessels. Aluminum vessels are eco-friendly compared to other materials. Moreover, they are easy in the sense of maintenance and repair, compared with other materials. Especially scrapped aluminum vessels are recyclable. Due to the tightening of international environmental regulations and a great upsurge of interest in resources recycling, the government has changed the material for small government vessels from FRP to aluminum alloys and there is an increase in the demand for aluminum vessels over time, centering on the domestic small- and medium-sized shipyards [1].

Welding is one of the most important factors in the production processes of aluminum vessels. In particular, welding of the aluminum ship body requires highly advanced skills, compared to the steel ship body and there is no smooth supply due to the extremely high personnel expenses [2]. During the subassembly phase of welding processes at the shipyard, stiffened plates are welded to add stiffness to horizontal panels. This time, different sizes and shapes of welding members used in the welding line make it hard to automatize the welding process and thus most small- and medium-sized shipyards are handling it manually [3].

S. M. Bae (✉) · W. J. Chung · H. G. Hwang
School of Mechatronics, Changwon National University, Changwon, Gyeongsangnam do,
South Korea
e-mail: wjchung@changwon.ac.kr

Y. J. Ahn
Robot Valley Corporation, Changwon, Gyeongsangnam do, South Korea

In this chapter, we aimed at structural stability and vibration reduction for high-quality welding of a three-axes Cartesian coordinate gantry robot devised for welding automation during subassembly of the ship body at a shipyard. In welding work, it is important to reduce vibration to negatively affect it. Structural instability and drive units can affect vibration [4]. Structural analysis will perform for parts that can cause structural instability due to load. Structural analysis will verify structural stability using ANSYS Workbench[®], based on the modeling of the gantry robot using Solidworks[®]. Also, the driving parts of the x axis and y axis are rack and pinion gear models, which have the disadvantage of vibration caused by backlash. A simulation to reduce vibration caused by backlash will be conducted using Recurdyn[®]. By using this simulation, we will evaluate the rack and pinion gear models how they affect the vibration of backlash.

2 Modeling of Three-Axes Cartesian Gantry Robot

A three-axes gantry robot is a tandem pulse MIG welding system for high-speed aluminum welding of different sizes of members during subassembly at a shipyard. Using Solidworks to make a three-dimensional modeling gantry robot for analysis is examined. This robot consists of six parts (x axis, y axis, z axis, platform, girder, and leg), and its main shape is shown in Figs. 1, 2, 3, 4, 5, 6, and 7.

The x axis is driven by the rack and pinion gears along the rail and there are wheels on both sides inside. This bears a load of nearly 6530 kg supporting all the parts excepting the rail. The shape of x axis is as shown in Fig. 8. The y axis is driven by the rack and pinion gears along the girder rail. One side is supported by the rollers and the other side is supported by the four LM (linear motion) guides. There are a controller for welding, a control panel, and a welding feeder in the first and second floors of the hand rail. The first floor and the second floor support a load of nearly 280 kg and 200 kg, respectively, and the shape of y axis is indicated as shown in Fig. 9. The z axis is driven by ball screws and supported by four LM guides. This bears a load of approximately 150 kg from the welding robot at the tip. The shape of z axis is shown in Fig. 10.

Fig. 1 The x axis shape of a three-axes gantry robot

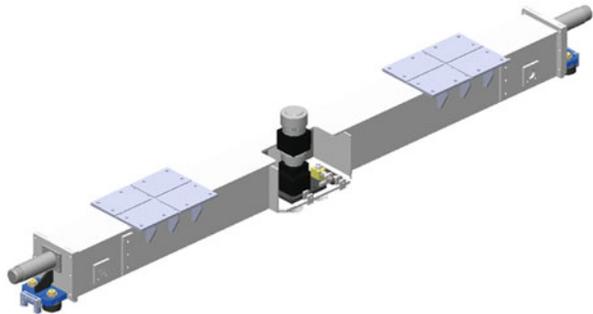


Fig. 2 The y axis shape of a three-axes gantry robot



Fig. 3 The z axis shape of a three-axes gantry robot

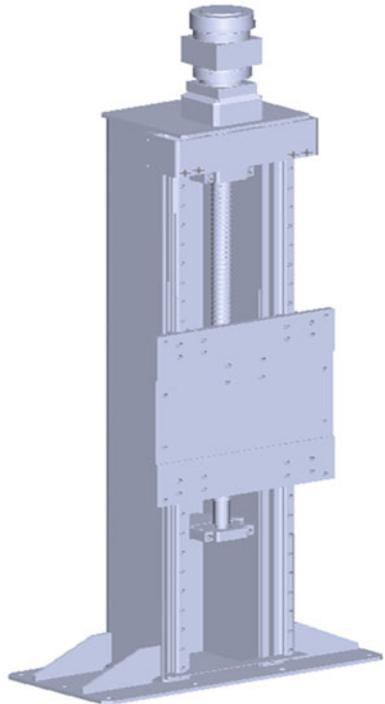


Fig. 4 The platform shape of a three-axes gantry robot

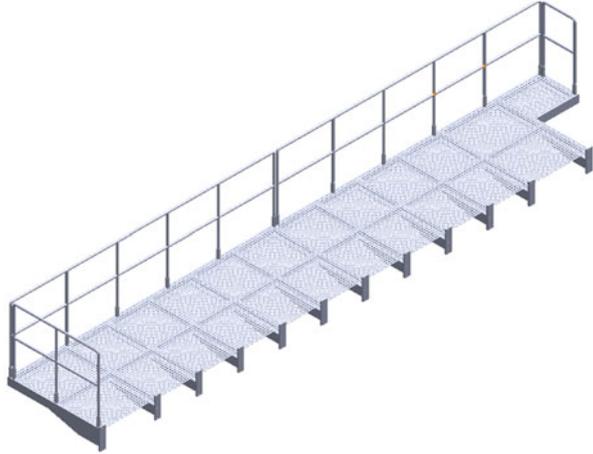


Fig. 5 The girder shape of a three-axes gantry robot

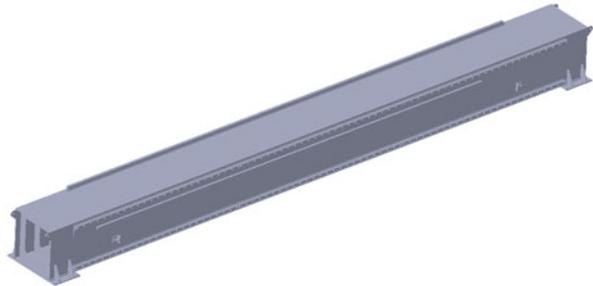
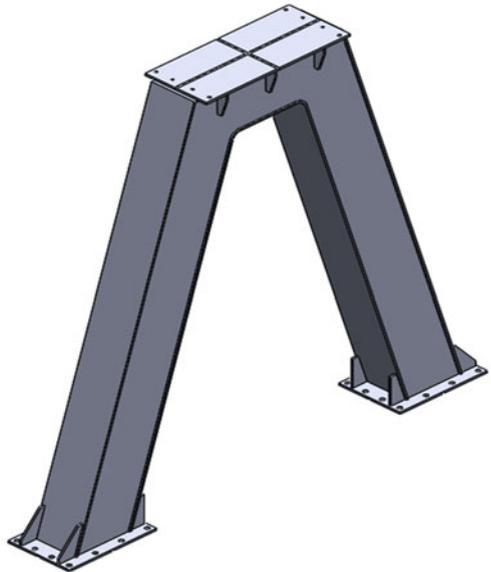


Fig. 6 The leg shape of a three-axes gantry robot



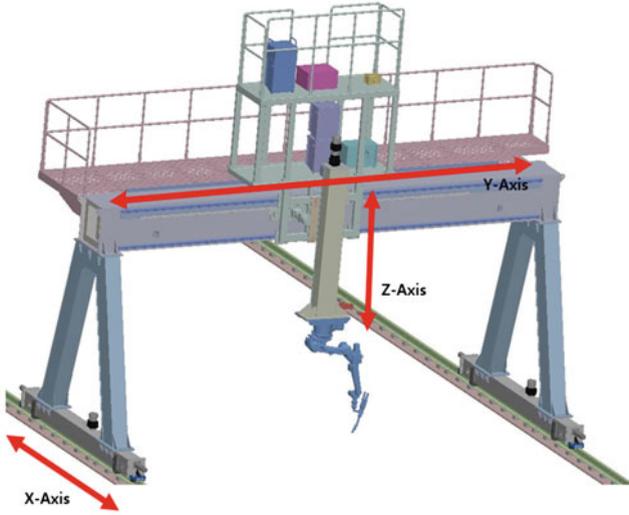


Fig. 7 The whole shape of a three-axes gantry robot

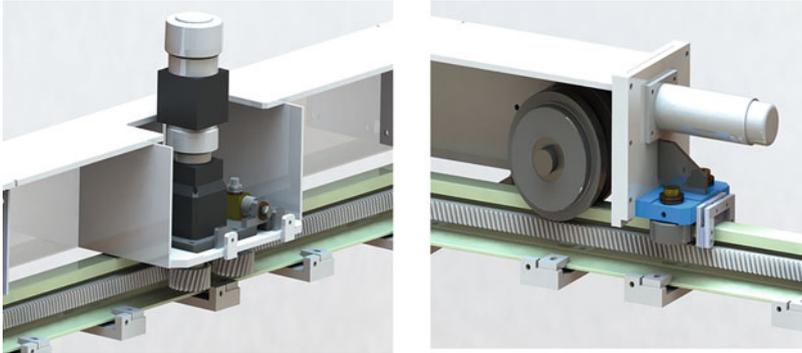
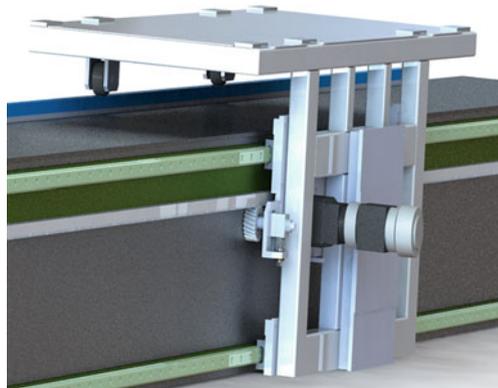


Fig. 8 The x axis driving part shape of a three-axes gantry robot

Fig. 9 The y axis driving part shape of a three-axes gantry robot



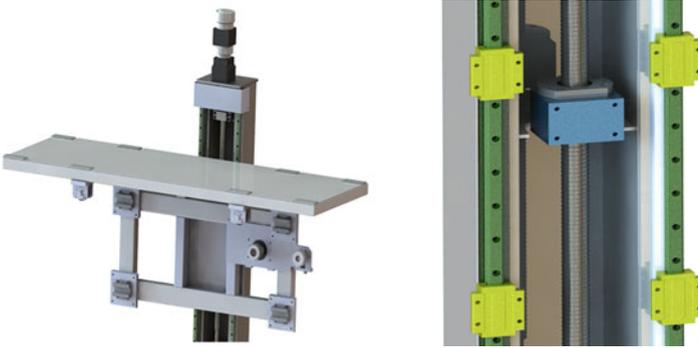


Fig. 10 The z axis driving part shape of a three-axis gantry robot

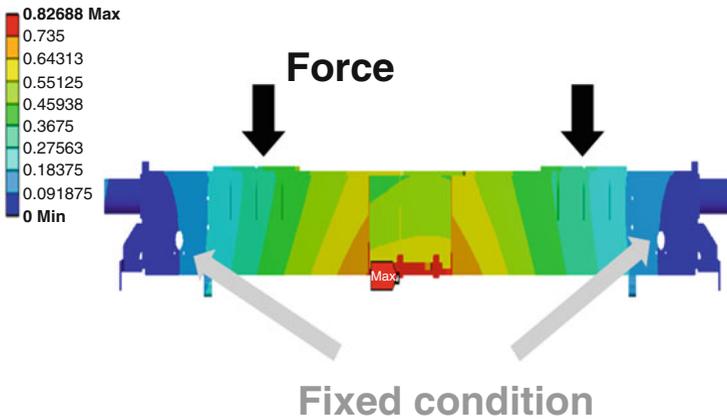


Fig. 11 Deformation distribution graph of x axis

3 Structural Stability Simulation of Gantry Robot

Structural instability negatively affects vibration which can reduce the quality of welding. It was verified by using ANSYS Workbench® to ensure that the gantry robot is structurally stable under the load it receives.

Structural analysis of x axis was conducted, because x axis might be structurally instable due to the load while supporting other parts excepting the rail. The gross weight of other parts is around 6530 kg. The load is applied to the side where the x axis supports the rest of the parts. Because there are four sides to support, the force is each 16,325 N. We assume that the axis of the wheel that touches the rail is fixed. The analysis result is shown in Figs. 11, 12, and 13.

The value of the maximum deformation is 0.826 mm. The point where it occurs is the center of the x axis frame. It tends to decrease as it goes out in relation to its center. This interpretation can be equivalent to the fact that *two concentric loads*

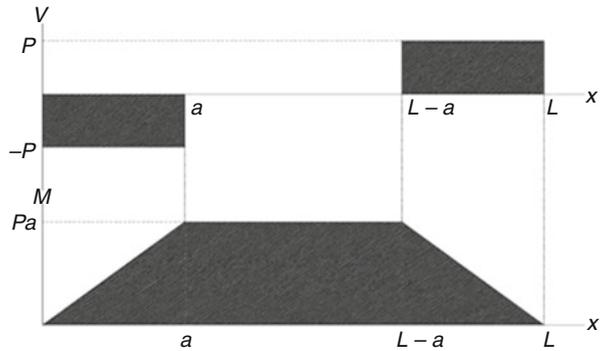


Fig. 12 Stress distribution graph of x axis



Fig. 13 Safety factor graph of x axis

Fig. 14 Shear force and bending moment of two equal interval concentrated loads for beam



with the same magnitude are applied on a fixed beam on both sides. The shear force and bending moment graph of the beam under *two* concentrated loads are shown in Fig. 14.

Thus, it is shown that the trend of simulation results is the same. The maximum stress value is 101.4 MPa and the minimum safety factor is 3.94, which occurs in the same position as the wheel axis. Since the minimum safety factor is higher than 2 (which can be considered as a standard safety factor), this axis can be considered as safe (Table 1).

Table 1 Structural analysis result of *x* axis for the gantry robot

Model	<i>x</i> axis
Max deformation	0.826 mm
Max stress	101.4 MPa
Min safety factor	3.94

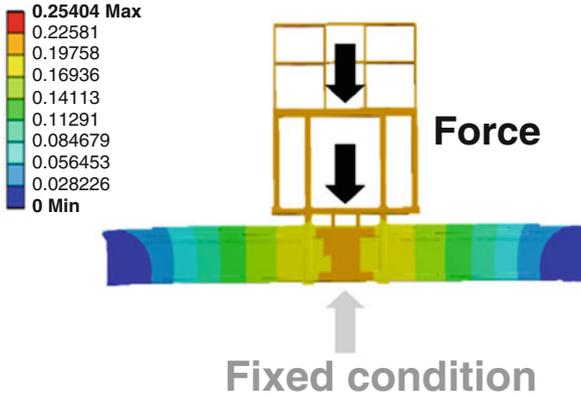


Fig. 15 Deformation distribution graph of *y* axis

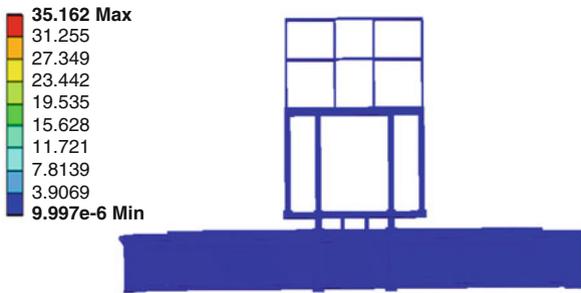


Fig. 16 Stress distribution graph of *y* axis

There are things for welding such as controllers in the hand rail of *y* axis. Since the load by these things may cause structural instability, structural analysis should be carried out. The first floor of handrail and the second floor of handrail support a weight of nearly 280 kg and nearly 200 kg, respectively. The distribution load of 2800 N was applied to the first floor, also 2000 N was applied to the second floor. The fixed plane is the floor of the *y* frame. The analysis result is shown in Figs. 15, 16, and 17.

The value of the maximum deformation is 0.254 mm. The point where it occurs is the center of the *y* axis frame. It tends to decrease as it goes out in relation to its center. This interpretation can be equivalent to the fact that *one concentrated load*

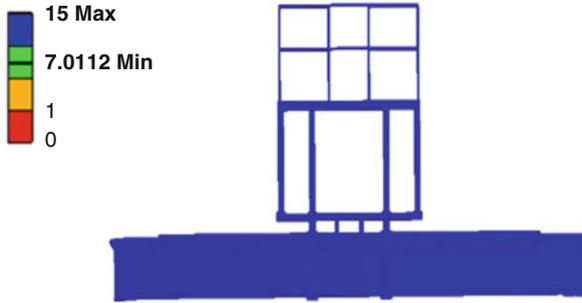


Fig. 17 Safety factor graph of y axis

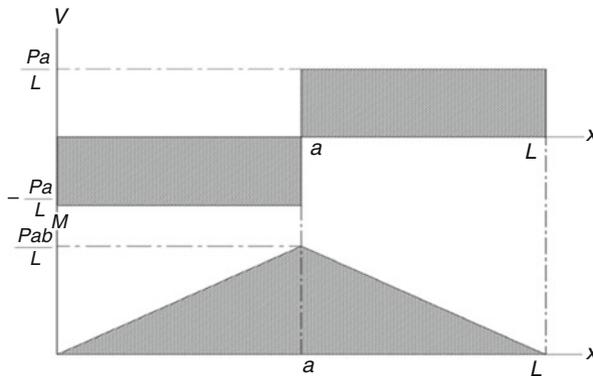


Fig. 18 Shear force and bending moment of one load for beam

Table 2 Structural analysis result of y axis of a gantry robot

Model	y axis
Max deformation	0.254 mm
Max stress	35.2 MPa
Min safety factor	7.1

is applied on a fixed beam. The shear force and bending moment graph of the beam under *one* concentrated loads are shown in Fig. 18.

Thus, it is shown that the trend of simulation results is the same. The maximum stress value is 35.2 MPa and the minimum safety factor is 7, which occur in the same position as the roller supporting the y axis from the girder. Since the minimum safety factor is higher than 2 (which can be considered as a standard safety factor), this axis can be considered as safe (Table 2).

Since z axis might be structurally instable due to the load that can be created during welding by a welding torch, structural analysis was conducted. The y axis is supported by the LM guide on one side and the roller on the other side. If the LM guide is not withstanding against load from welding operation, the rail deviation of roller can be caused. The welding torch weighed about 150 kg and structural

analysis targeted three cases of acceleration: 2 m/s^2 , 4 m/s^2 , and 6 m/s^2 . The fixed plane is the LM guide added plate. The analysis result is shown in Figs. 19, 20, 21, and 22.

The value of the maximum deformation is 0.032 mm, 0.1 mm, and 0.138 mm at each acceleration of 2 m/s^2 , 4 m/s^2 , and 6 m/s^2 , respectively. The point where it occurs is the center of the z axis frame. It tends to decrease as it goes out in

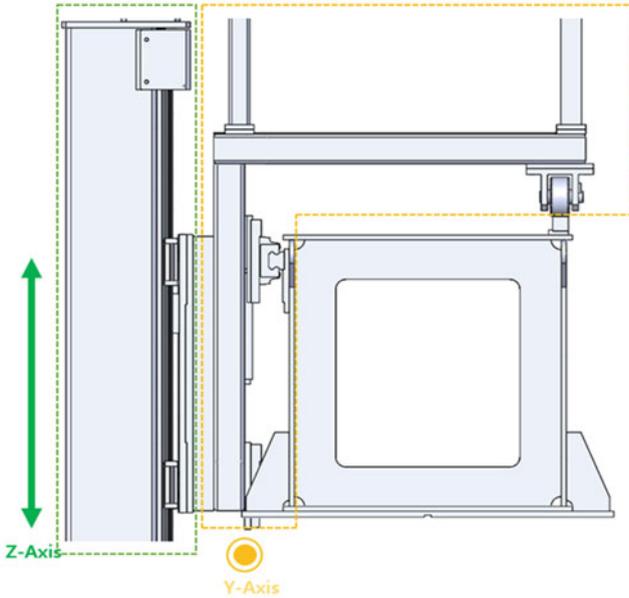


Fig. 19 The y, z axis shape of a three-axes gantry robot

Fig. 20 Deformation distribution graph of z axis

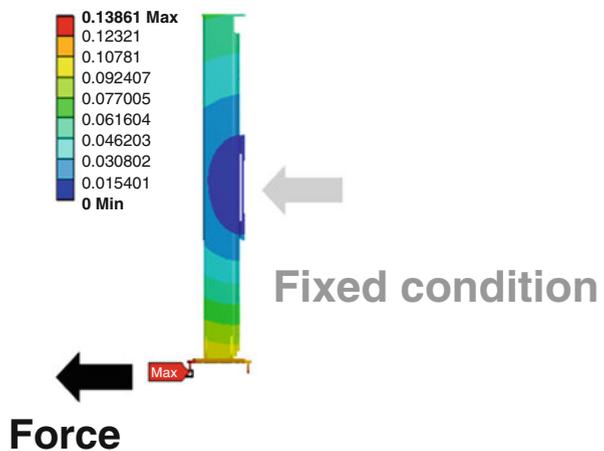
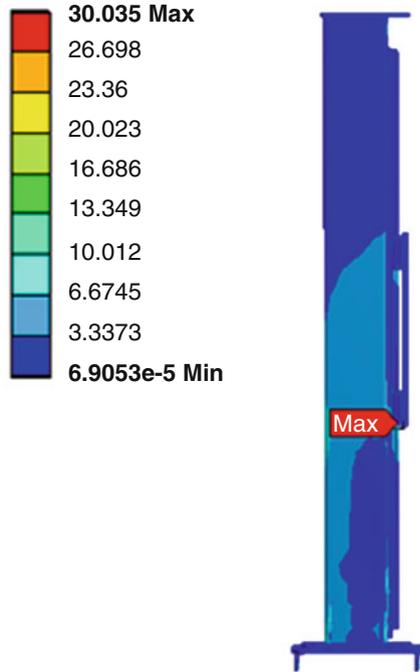


Fig. 21 Stress distribution graph of z axis



relation to its center. The tendency of z axis structural analysis is the same as that of y axis structural analysis (which is shown in Fig. 18). The maximum stress was 30 MPa at 6 m/s² that was the biggest acceleration among the three cases and this was generated in the LM guide. The minimum safety factor revealed in the LM guide was 8.23 that was higher than the general safety factor, 2 (which can be considered as a standard safety factor). Besides, the real acceleration of the welding torch during operation did not exceed 6 m/s², which means this axis is safe (Table 3).

From the viewpoint of the fact that structural instability can negatively affect vibration, the structural analysis was conducted so that all of x, y, and z axes were regarded as safe structurally.

4 Verification for Vibration Reduction of Gantry Robots

The x and y axes of a gantry robot are driven by the rack and pinion gears. One of the drawbacks of the rack and pinion gear is vibration due to backlash. A simulation was conducted in order to investigate a know-how that could reduce the vibration of rack and pinion gear. The simulation was conducted on how *adding pinion gears* would affect vibration. Rail, rack gear, pinion gear, and wheel in the truck are the same size as the actual model, and the parts that are not needed to drive were simplified. The

Fig. 22 Safety factor graph of z axis

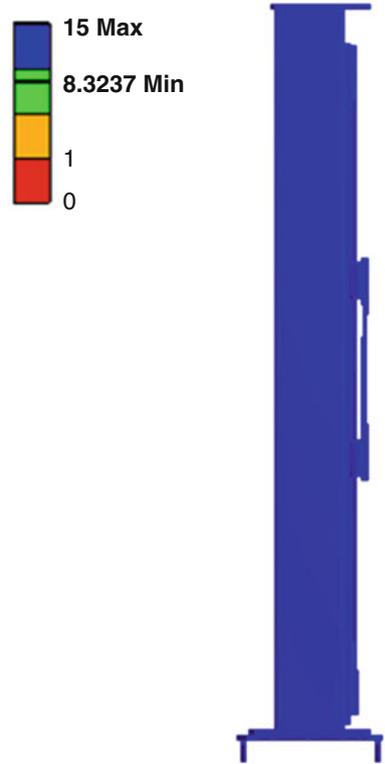


Table 3 Structural analysis result of z axis for the gantry robot

	Case1 (2 m/s ²)	Case2 (4 m/s ²)	Case3 (6 m/s ²)
Max deformation	0.032 mm	0.1 mm	0.138 mm
Max stress	4.5 MPa	21.7 MPa	30 MPa
Min safety factor	15	11.5	8.23

motor hole in the model with one pinion gear was placed right in the center of the plate and the hole of the model with two pinion gears was placed that it is equally spaced by both sides of the wall. A rotation speed of 15 mm/s was applied to each pinion gear and a gravity was applied in the z axis direction. The total analysis time was set to 3 seconds and the direction was changed at about 1.5 seconds (Figs. 23 and 24).

Vibration of the x axis and y axis is inevitable when driving. So the degree of vibration of the z axis was the basis of simulation. A velocity graph in the direction of the z axis of the driving truck was extracted as the simulation result of Recurdyn[®] to determine the vibration of the model and the results are shown in Figs. 25 and 26.

Usually, the maximum vibration (in this case, the maximum velocity) will occur due to backlash. Table 4 illustrates that increasing the number of pinion gear can

Fig. 23 One pinion gear simulation model

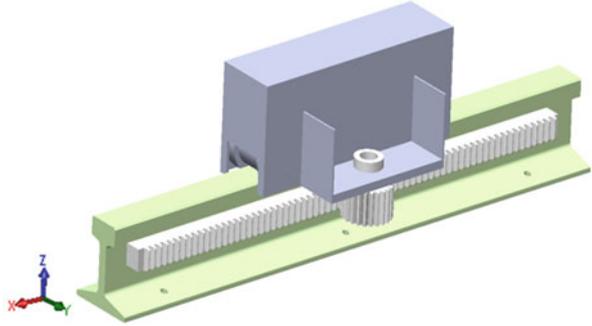


Fig. 24 Two pinion gears simulation model

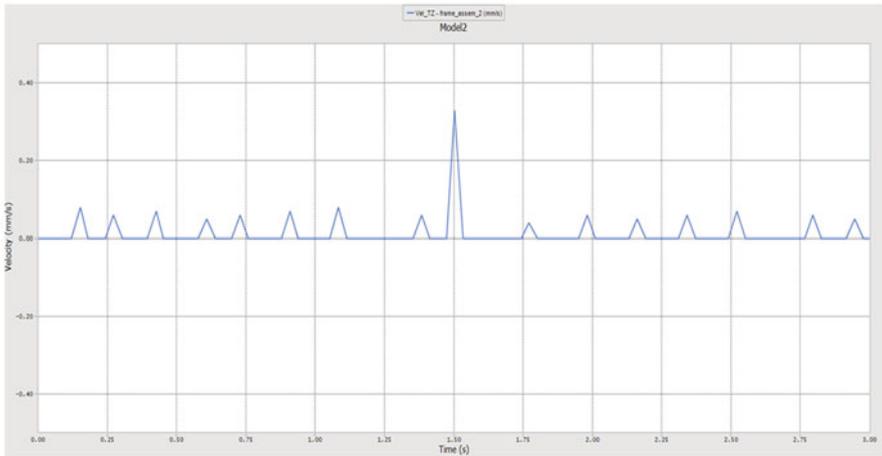
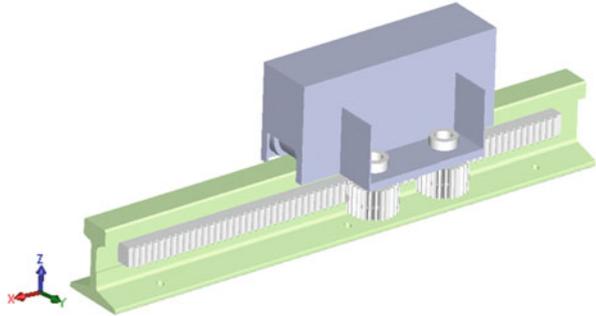


Fig. 25 One pinion gear simulation by using Recurdyn®

reduce the maximum vibration (velocity) caused by backlash but average vibration and minimum vibration is increased in a tolerable range Therefore, *adding two pinion gears to the rack is verified to be a better strategy in reducing vibration rather than adding only one pinion gear to the rack.*

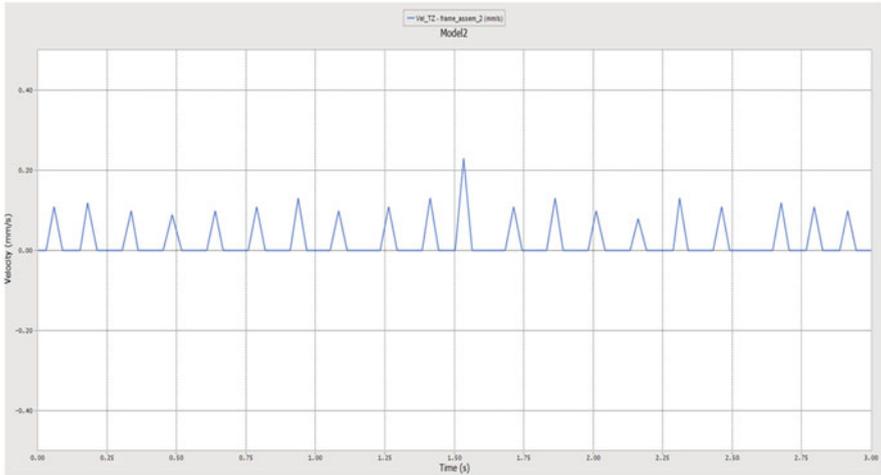


Fig. 26 Two pinion gears simulation by using Recurdyn®

Table 4 The results of dynamics analysis for each simulation model

z axis	One pinion model	Two pinion model
Maximum velocity	0.33 mm/s	0.23 mm/s
Average velocity	0.12 mm/s	0.15 mm/s
Minimum velocity	0.08 mm/s	0.1 mm/s

5 Conclusion

In this chapter, we aimed on structural stability and vibration reduction for high-quality welding of a three-axes Cartesian coordinate gantry robot devised for welding automation during subassembly of the ship body at a shipyard. First, structural analysis has been performed for parts that can cause structural instability due to load. Structural analysis has verified structural stability using ANSYS Workbench®, based on the modeling of the gantry robot using Solidworks®. As a result of structural simulation, this gantry robot model was structurally safe. Also, the driving parts of the *x* axis and *y* axis are rack and pinion gear models, which have the disadvantage of vibration caused by backlash. A simulation to reduce vibration caused by backlash has been conducted using Recurdyn®. By using this simulation, we have evaluated the rack and pinion gear models how they affect the vibration of backlash. Dynamic analysis confirmed that adding pinion gears can reduce maximum vibration due to backlash, but increase average and minimum vibration. The minimum and average vibration is negligible in welding work, so using two pinion gears will improve the quality of the welding.

Acknowledgment This study was conducted as part of the “Development of High Efficiency MIG Welding Robot System for Aluminum Ships” (Project No. 20005862), which has been supported by the Ministry of Trade, Industry and Energy in the Republic of Korea.

References

1. S.J. Kim, S.K. Jang, M.S. Han, Evaluation of mechanical characteristic of Al alloy of ship's welded with various welding techniques. *J. Korean Soc. Mar. Environ. Saf.* **13**(3), 223–228 (2007)
2. J.H. Lee, S.Y. Hwang, Review on the manufacturing and assembly technology of aluminum ship construction. *J. Korea Ship Saf. Technol. Authority* **32**(2012), 2–11 (2012)
3. Y.B. Woo, M.S. Han, Characteristics evaluation on dissimilar metal welding of Al ship. *J. Korean Soc. Navigation* **2008**(3), 15–16 (2008)
4. J.C. Cahng, J.P. Dong, A study on dynamic modeling and vibration analysis of gantry robot. *J. Korean Soc. Indust. Appl.* **17**(4), 211–216 (2014)

Long Short-Term Memory Neural Network on the Trajectory Computing of Direct Dynamics Simulation



Fred Wu, Tejaswi Jonnalagadda, Colmenares-diaz Eduardo, Sailaja Peruka, Poojitha Chapala, and Pooja Sonmale

1 Introduction

Classical trajectory chemical dynamics simulation is a widely and powerful tool that has been used to study reaction dynamics since the 1960s [1]. In contrast to the variational transition state theory (VTST) and reaction path Hamiltonian methods [2], they provide much greater insight into the dynamics of reactions for the classical equations of motion of the atoms that are numerically integrated on a potential energy surface (PES). The traditional approach uses an analytic function that is gotten by fitting ab initio and/or experimental data [3] to construct the surface. With regard to a small number of atoms or a high degree of symmetry [4, 5], it is practical. Researchers recently proposed additional approaches and algorithms for representing PESs. Wang and Karplus firstly demonstrated that the trajectories may be integrated “on the fly” when the potential energy and gradient are available at each point of the numerical integration according to an electronic structure theory calculation. During the numerical integration, the method directly calculates the local potential and gradient under an electronic structure theory in a “direct dynamics” simulation. However, regarding a high-level electronic structure theory, the computation of direct dynamics simulations becomes quite expensive. Thus, it is important to use the largest numerical integration step size when maintaining

F. Wu (✉) · T. Jonnalagadda · S. Peruka · P. Chapala · P. Sonmale
Department of Mathematics and Computer Science, West Virginia State University, Institute,
WV, USA

e-mail: heng.wu@wvstateu.edu; tjonnalagadda@wvstateu.edu; speruka@wvstateu.edu;
pchapala@wvstateu.edu; psonmale@wvstateu.edu

C.-d. Eduardo
Department of Computer Science, Midwestern State University, Wichita Falls, TX, USA
e-mail: eduardo.colmenares-diaz@msutexas.edu

the accuracy of the trajectory. In order to use a larger integration step, Helgaker et al. adopt the second derivative of the potential (Hessian). After the Hessians are gotten directly by an electronic structure theory, using a second-order Taylor expansion, a local approximation PES can be constructed and the trajectories can be approximately calculated. For local quadratic potential is only valid in a small region (named a “trust radius”), the equations of motion are only integrated under the trust radius. The new potential, gradient, and Hessian, calculated again at the end of the trust radius, define a new local quadratic PES where the integration of the equations of motion is successive. Millam et al. used a fifth-order polynomial or a rational function to fit the potential between the potential, gradients, and Hessians at the beginning and end of each integration step. It provides a more accurate trajectory in the trust region and calculates a larger integration steps. That involves a predictor step, the integration on the approximate quadratic model potential. The following step, the fitting on the fifth order PES between the starting point and the end point in the trust radius, is also called the “corrector step.” It is named the Hessian-based predictor-corrector integration scheme. Around it, some scholars proposed their own methods [6]. Because of extrapolation, errors in prediction-correction algorithms grow rapidly; usually four predictions are followed by an ab initio calculation. This limits the improvement of computing performance.

The successful application of the prediction of deep learning in computational chemistry greatly expanded its application. Deep learning is a machine learning algorithm, not unlike those already in use in various applications in computational chemistry, from computer-aided drug design to materials property prediction [7, 8]. Deep learning models achieved top positions in the Tox21 toxicity prediction challenge issued by NIH in 2014 [9]. Among some of its more high-profile achievements include the Merck activity prediction challenge in 2012, where a deep neural network not only won the competition and outperformed Merck’s internal baseline model but also did so without having a single chemist or biologist in their team. Machine learning (ML) models also can be used to infer quantum mechanical (QM) expectation values of molecules, based on reference calculations across chemical space [10]. Such models can speed up predictions by several orders of magnitude, demonstrated for relevant molecular properties such as polarizabilities, electron correlation, and electronic excitations [11]. LSTM is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. The prediction of LSTM has been widely used in different fields [12, 13].

In this paper, we explore the idea of integrating LSTM layer with chemistry dynamics simulations to enhance the performance in trust radius. This idea is inspired by the recent development and use of LSTM in material simulations and scientific software applications [14]. We employ a particular example, CO₂ molecular dynamics simulation on NWChem/Venus (cdssim.chem.ttu.edu) package [15], to illustrate this idea. LSTM has been used to predict the energy, location, and Hessian of atoms. The results demonstrate that LSTM-based memory model, trained on data generated via these simulations, successfully learns pre-identified key features associated with the energy, location, and Hessian of molecular system. The deep learning approach entirely bypasses simulations and generates predictions

that are in excellent agreement with results obtained from explicit chemistry dynamics simulations. The results demonstrate that the performance gains of chemical computing can be enhanced using data-driven approaches such as deep learning which improves the usability of the simulation framework by enabling real-time engagement and anytime access.

This paper is organized as follows. Section 2 presents the idea that integrates chemistry dynamics simulations with LSTM. Section 3 shows the experiment setting and results on CO₂ molecular dynamics simulation, followed by data analysis. Section 4 presents the conclusions and lays out future work.

2 Methodology

2.1 Prediction-Correction Algorithm

In chemistry dynamics simulation, Hessian's calculation consumes most of the CPU time because Hessian is the third derivative of the position. Hessian updating is a technique frequently used to replace electronic structure calculations of the Hessian in optimization and dynamics simulations. Existing generally applicable Hessian update schemes, e.g., the symmetric rank one (SR1) scheme, Powell-symmetric-Broyden (PSB) method, the scheme of Bofill, the Broyden-Fletcher-Goldfarb-Shanno (BFGS) scheme, the scheme of Farkas and Schlegel, and other Hessian update schemes, are based on the Eq. (1).

$$H(X_{k+1})(X_{k+1}-X_k) = G(X_{k+1}) - G(X_k) \quad (1)$$

where $G(X)$ and $H(i)$, respectively, denote the gradient and Hessian of the potential energy at point X . Some researchers employed Hessian update method to build Hessian-based prediction-correction integration method to calculate the trajectory of atom in order to reduce the calculation time of Hessian and ab initio.

As illustrated in Fig. 1, in each time step of the integration method, the prediction is used to identify the direction of the trajectory; ab initio potential energy, ab initio gradient, and ab initio or Hessian are computed at the end point $X_{i,p}$ of the predicted trajectory. The potential information calculated at the end of predicted trajectory is used with the potential energy information at point $X_{i-1,p}$ near the trajectory starting point X_{i-1} of this time step, which is the end point of corrected trajectory of the previous time step, to interpolate a highly accurate local PES. This highly accurate PES is used in the correction phase of the time step to re-compute a more accurate trajectory.

In each time step, to obtain an accurate predicted trajectory, the prediction utilizes the Hessian in addition to the potential energy and its gradient. Assuming the current time step is the i th time step, the potential energy information needed during the prediction to integrate the trajectory is obtained by the quadratic expansion.

$$E(X) = E(X_{i-1,p}) + G(X_{i-1,p})(X - X_{i-1,p}) + 1/2(X - X_{i-1,p})^T H(X_{i-1,p})(X - X_{i-1,p}) \quad i > 2 \quad (2)$$

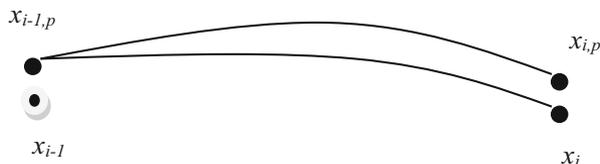


Fig. 1 During the i^{th} step, the algorithm first predicts the trajectory from X_{i-1} to $X_{i,p}$ using potential approximated by the quadratic Taylor expansion about $X_{i-1,p}$ and then performs electronic structure calculation of the potential energy information at $X_{i,p}$ and re-integrate the trajectory from X_{i-1} to X_i using potential interpolated from ab initio potential information at $X_{i-1,p}$ and $X_{i,p}$

P is an integer. About the point $X_{i-1,p}$, the end point of the predicted trajectory of the $(i-1)^{\text{th}}$ time step at which ab initio potential energy $E(X_{i-1,p})$, ab initio gradient $G(X_{i-1,p})$, and ab initio or updated Hessian $H(X_{i-1,p})$ have been computed on a region within a trust radius from $X_{i-1,p}$.

If we use $X_{i-1,p}$ as the current location, the next part will show how to calculate the potential energy for the next X location. We can calculate the potential energy (P) and gradient (G) at the $X_{i-1,p}$ from known position. For example, there are eight atoms in $\text{F}^- + \text{CH}_3\text{OOH}$. There are $3 \times N$ dimensions in the gradient and location vectors and N^2 dimensions in the Hessian matrix of the reaction system. Therefore, most of the calculation of Eq. (2) is to compute $H(X_{i-1,p})$. The biggest challenge is to choose different approaches to fast the calculation of $H(X_{i-1,p})$ with the position, and others of the current location, at the same time, cannot enlarge the system error.

2.2 Long Short-Term Memory

As shown in Fig. 2, a neural network is the connection of many single neurons, and an output of a neuron can be an input of another neuron. Each single neuron has an activation function. The left layer of the neural network is called the input layer, and it includes X1, X2, X3, X4; the right layer of it is the output layer, and it involves Z1. The other layer is the hidden layer, and it covers Y1, Y2, Y3.

Recurrent neural network (RNN) is a typical kind of neural network, as shown in the leftmost part of Fig. 3.

Like the leftmost of Fig. 3, RNN is a neural network containing loops. N is a node of neural network. I stands for input and O for output. Loops allow information to be transmitted from the current step to the next step. RNN can be regarded as a multiple assignment of the same neural network, and each neural network module transmits the message to the next one. The right side of Fig. 3 corresponds to the unfolding of the left side. The chain feature of RNN reveals that RNN is essentially related to sequences and lists. RNN applications have been successful in speech recognition, language modeling, translation, and picture description, and this list is still growing. One of the key features of RNN is that they can be used to transmit the previous information to the current task. But the distance from previous step to related step is not too long.

Fig. 2 The structure of a neural network

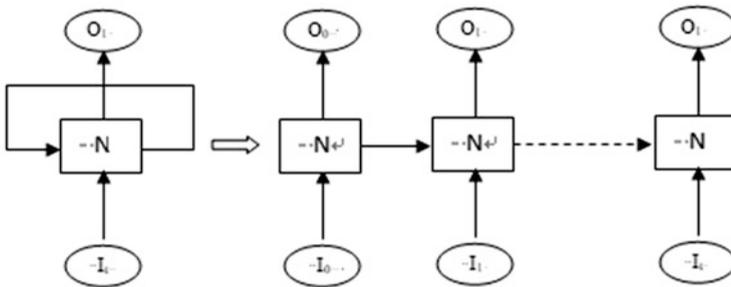
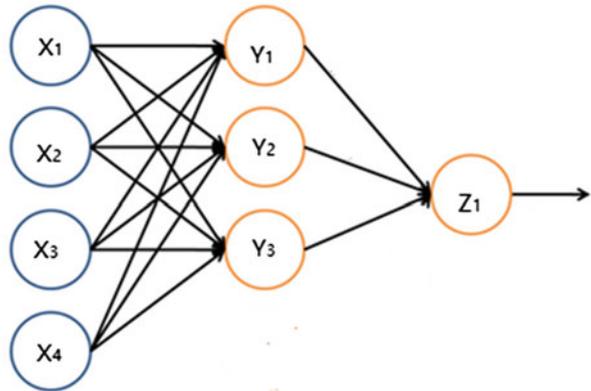


Fig. 3 The structure of a recurrent neural network and its unfolding

LSTM (long short-term memory) overcomes this shortcoming. LSTM is a special type of RNN. LSTM solves the problem of long-term dependence of information. LSTM avoids long-term dependencies through deliberate design. Figure 4 shows the structure of a node in LSTM, where a forget gate can be observed. The output of the forget gate is “1” or “0”; “1” means full reserve, and “0” is abandon completely. The forget gate determines which information will be retained and what will be discarded. The upper horizontal line allows the input information cross neutral node without changing in Fig. 4. There are two types of gates in a LSTM node (input and output gates). The middle gate is an input gate, which determines the information to be saved in the natural node. F means function modular and create a new candidate value vector. The right gate is the output gate. The F module closed to the output gate determines which information of the natural node will be transmitted to the output gate.

A node has three gates and a cell unit as shown in Fig. 4. The gates use sigmoid as activation function; the tanh function is used to transfer from input to cell states. The following are to define a node. For the gates, the function are

$$i_t = g(w_{xi}x_t + w_{hi}h_{t-1} + b_i) \tag{3}$$

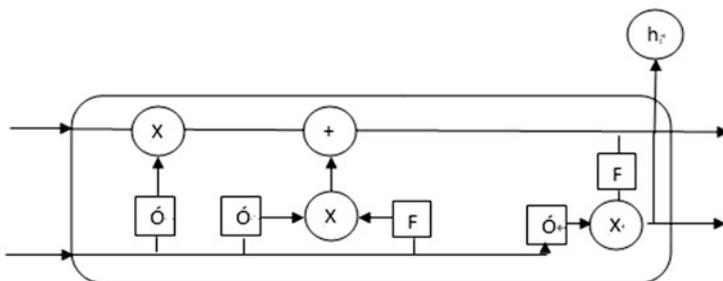


Fig. 4 The structure of a LSTM node

$$f_t = g(w_{xf}x_t + w_{hf}h_{t-1} + b_f) \quad (4)$$

$$f_o = g(w_{xo}x_t + w_{of}h_{t-1} + b_o) \quad (5)$$

The transfer for input status is

$$c_in_t = \tan h(w_{xc}x_t + w_{hc}h_{t-1} + b_{o_in}) \quad (6)$$

The status is updated by

$$c_t = f_t * c_{t-1} + i_x * c_{in_t} \quad (7)$$

$$h_t = o_t * \tan h(c_t) \quad (8)$$

The workflow of a node is shown in Fig. 5, and Fig. 6 shows the flowchart for LSTM.

2.3 Model

The calculation of position of the atom, the energy of the system, and Hessian occupies almost all the CPU time in chemistry dynamics simulations. Figure 7 illustrates the Hessian-based predictor-corrector algorithm in chemistry dynamics simulations. At each time step, the potential energy, kinetic energy, velocity, Hessian, and other parameters are calculated from the position of the atom. In Fig. 1, assuming $X_{i-1,p}$ is the current point, the calculation potential energy of next point X is as follows. The gradient and potential energy of the current point can be calculated from the known location of the point. Assuming eight atoms, the dimension of gradient and location will be $3 \times N$, which of H . will be N^2 . Hence, the largest calculation of Eq. 2 will be to calculate $H(X_{i-1,p})$. It is the focus of the study of various algorithms to quickly and accurately calculate.

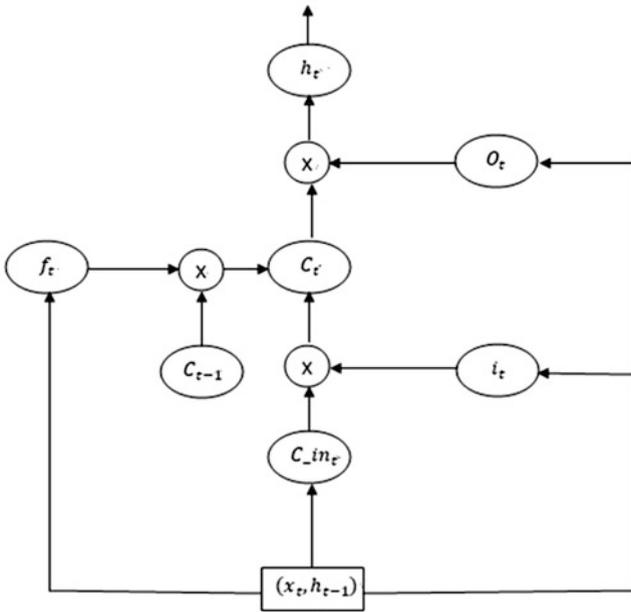


Fig. 5 The workflow of status changing of neutral node

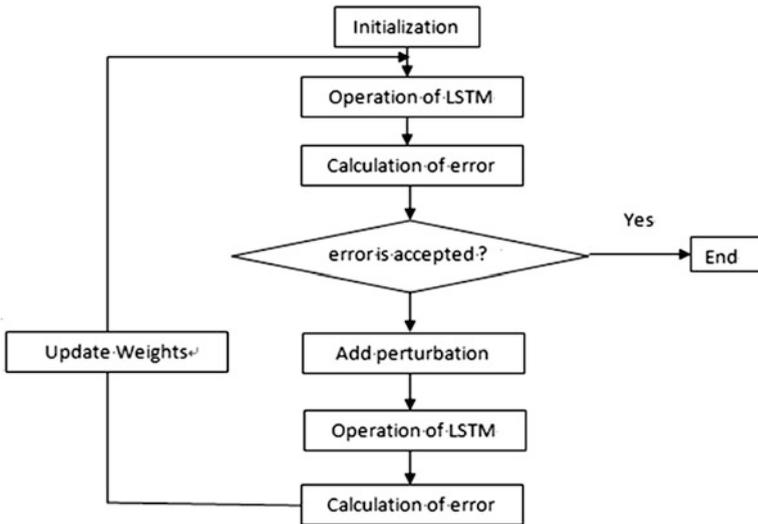
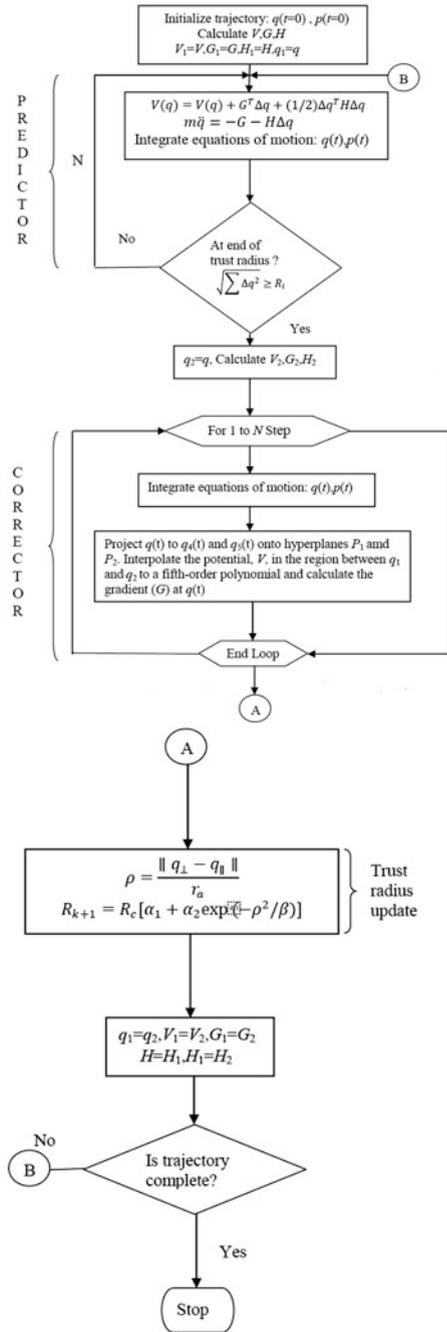


Fig. 6 The flowchart of LSTM

Fig. 7 Flowchart representation of the complete Hessian-based predictor-corrector integrator



Algorithm 1

```
Input: Atomic initial parameters
Parameter: location of atoms, initial energy info
Output: the trajectory of atoms
1.   ab initio computing
2.   while less than steps do
3.     while less than training step do
4.       exec Predictor-Corrector
5.       train deep learning model
6.     end while
7.     predict the location, energy, and Hessian
8.     output trajectory
9.   end while
10. return
```

$H(X_{i-1}, p)$ according to the location and time information of the current point, simultaneously systematic error is least required. Researchers proposed some Hessian update methods to saving computing time [6]. Deep learning will be used to predict the location, energy, and Hessian of atoms. Therefore deep learning will be used three times instead of predictor-corrector. It is important to understand that our deep learning model needs to be trained and initialized before predicting. The result of this approach is a novel predictor-corrector algorithm with deep learning.

3 Experimental Results

To test the algorithm with deep learning, we implemented the integration algorithm in the VENUS (cdssim.chem.ttu.edu) dynamics simulation package interfaced with the electronic structure calculation NWChem [15]. We chose the reaction system CO_2 as our testing problem. In the tests, ab initio potential energy, gradient, and Hessian were calculated using the density function theory 6-311 + G**, and ab initio Hessian is calculated once in every five steps during training. In the remaining nine steps, the new update scheme is used. We calculated a trajectory for the chemical reaction system with 500×0.67 integration steps, where each step has a fixed size of 0.02418884 fs (100 a.u.; 1 a.u. = 2.418884×10^{-17} s). The remaining 500×0.33 steps were predicted by the proposed deep learning algorithm. There are three prediction parameters in our test. They are atomic position, energy, and Hessian, respectively.

Figure 8 illustrates the computational energy and its predicted values. The above is the CO_2 system computational energy chart. The horizontal coordinate is the time step and the vertical one is the energy value. The yellow region represents training data and green section predicted values. After more than 300 training steps,

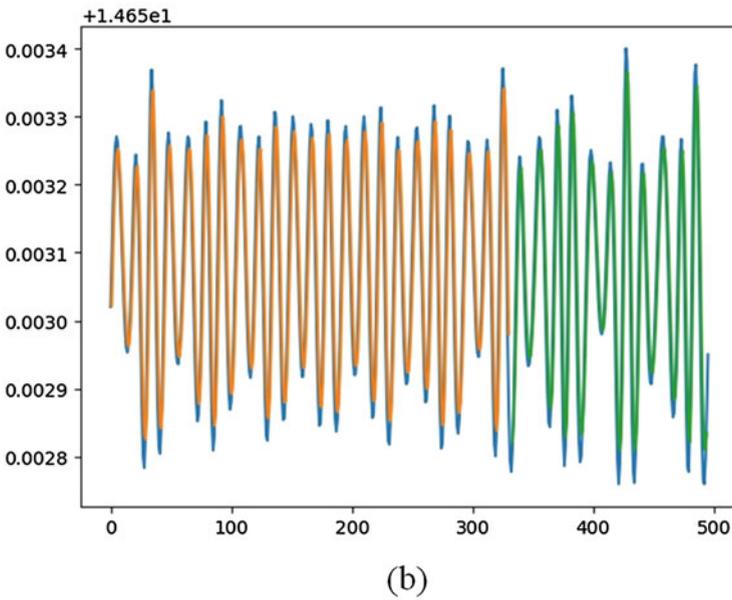
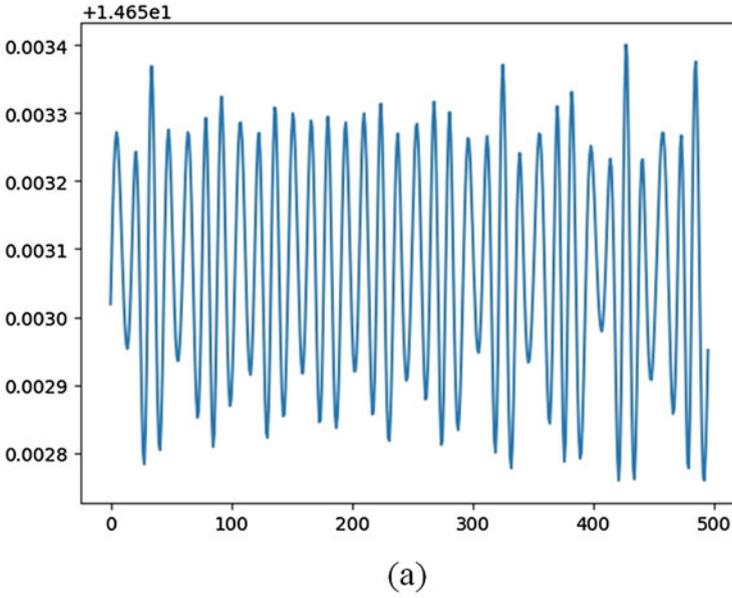


Fig. 8 The energy of CO₂ system: (a) output of the prediction-correction algorithm. (b) The yellow region corresponds to training data, and green is prediction data

Table 1 Relative error between system energy prediction and computational value (500 steps)

Computational data (D1)	Predicted data D2	$ (D1-D2) / D1 $ (%)
14.653119	14.65316	0.1%
14.653076	14.653113	<0.01%
14.653034	14.653073	0.1%
14.653004	14.653036	0.15%
14.652985	14.653007	<0.01%

the predicted value is almost the same as the calculated values. Table 1 lists some relative errors. We find the relative error to be less than 0.15%.

Figure 9 shows a carbon atomic location chat. The above is the computational values, and the following is the training and predicted values. The horizontal coordinate is the time step and the vertical one is the atomic location. Table 2 is some relative error between predicted and computational values. We find the relative error less than 13%. While for 5000 steps test, the error reduces 8% and even 1% in Table 3.

Figure 10 is one of the Hessian charts with 500 steps. Figure 11 is one of the Hessian charts with 5000 steps. The above is the computational values, and the following is the training and predicted values. The horizontal coordinate is the time step, and the vertical one is the Hessian value. Table 4 is some relative error between predicted and computational values. We find that the minimum relative error is 3.3% and some even over 7%. Although it is 500 steps, Hessian calculated only 100 steps because of the predictive-correction algorithm. Therefore, the size of the training set is less than 67, and the relative error is relatively large.

The prediction-correction algorithm can reduce CO₂ reaction system dynamics simulation time from months to days. The stability of the prediction-correction algorithm becomes very weak as simulation goes on. In addition, there must be an ab initio calculation every few steps in the prediction-correction algorithm. As the prediction step increases, the stability becomes weaker. Deep learning can reduce the simulation time of the reaction system by one-third. The prediction step can reach over 120 steps without affecting the system error after enough training. If some reinforcement learning and other methods are used, the calculation time will be further reduced and the prediction steps will be more.

4 Conclusion and Future Work

From the above discussion, the error has been reduced to 0.15% in 500 steps for calculation parameters with a small changing. For parameters with a large changing, like the trajectory of an atom, the prediction accuracy can only be improved by increasing the number of trainings, such as from 500 to 5000.

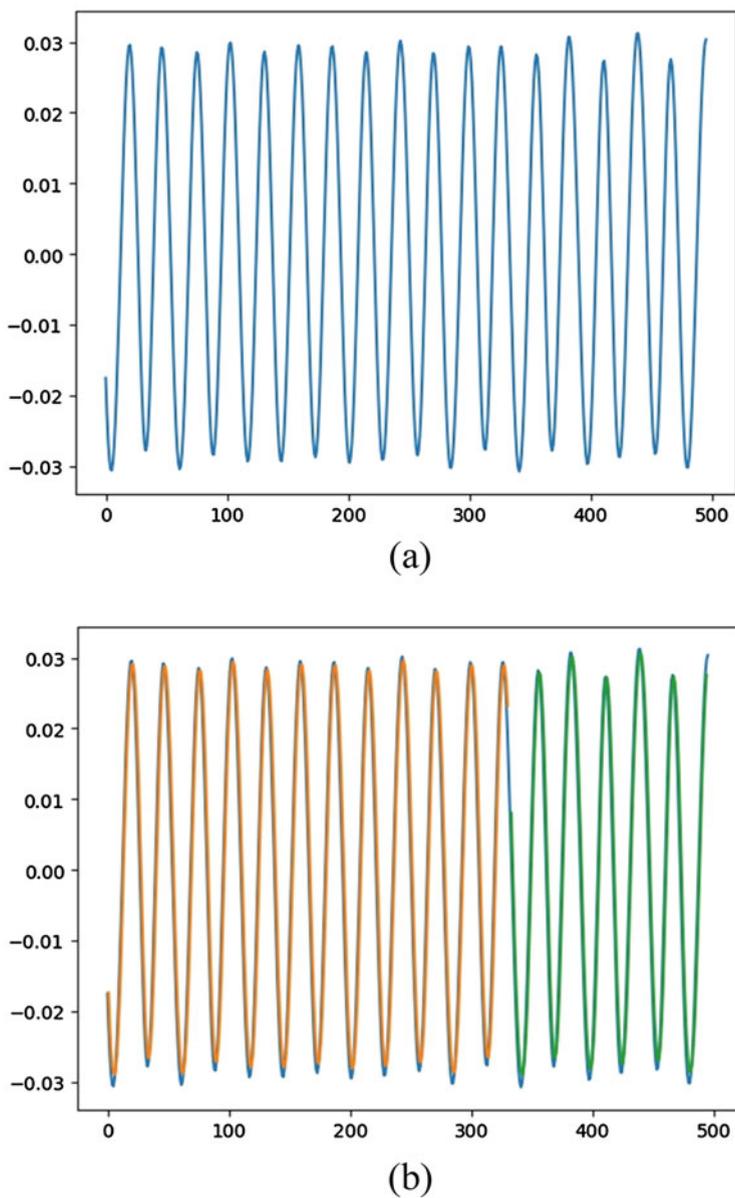


Fig. 9 The location of atoms in CO₂ system: (a) output for the prediction-correction algorithm. (b) The yellow region corresponds to training data, and green represents prediction data

Table 2 Relative error between atomic position prediction and computational value (500 steps)

Computational data (D1)	Predicted data D2	$ (D1-D2) / D1 $ (%)
-0.024813	-0.026670	7.5%
-0.020561	-0.024086	17%
0.021499	0.021833	20%
0.025055	0.017207	12.8%
0.027043	0.025103	7%

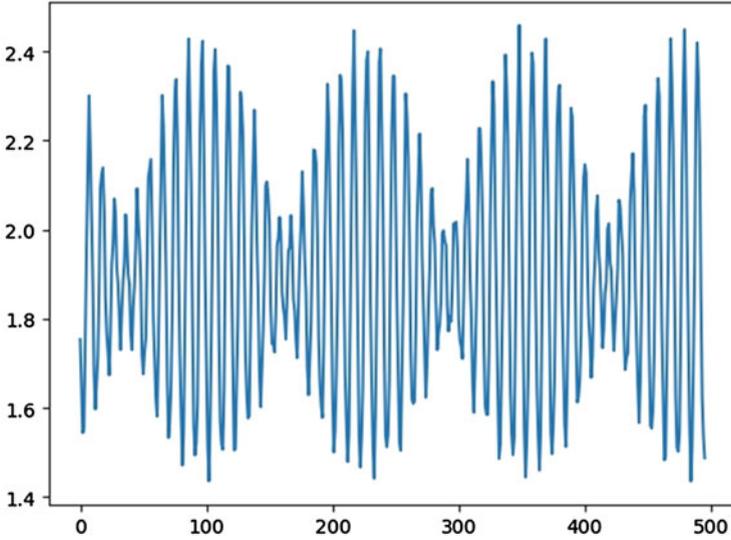
Table 3 Relative error between atomic position prediction and computational value (5000 steps)

Computational data (D1)	Predicted data D2	$ (D1-D2) / D1 $ (%)
0.02648124	0.0282837	6%
0.02836987	0.0286955	1%
0.0288303	0.0277771	3.6%
0.02780569	0.0255204	8%

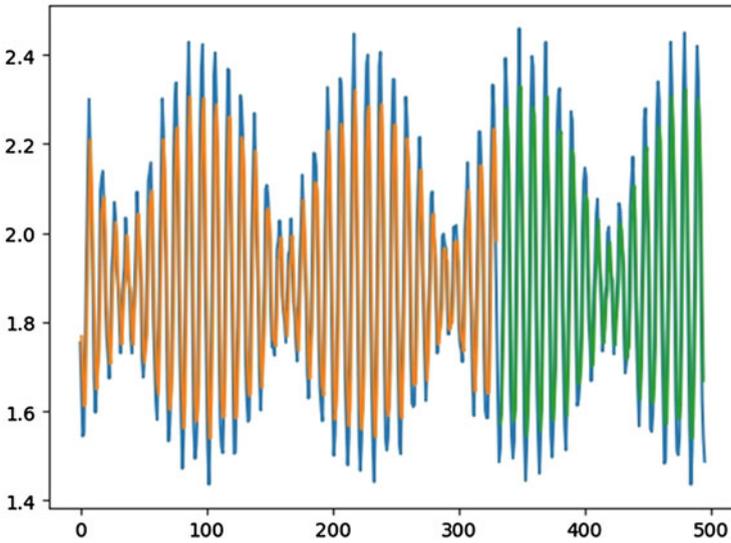
In this paper, a new molecular dynamics simulation algorithm is proposed by combining deep learning and predictive correction algorithms. The new algorithm can reduce the calculation time of the system by one-third without increasing the error. In the future, the enhanced learning and parameter migration will be used to further reduce the calculation time. Then monodromy matrix [16–18] will be used to monitor the change of the calculation error.

5 Acknowledgment

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g.” Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.



(a)



(b)

Fig. 10 The Hessian of CO₂ system with 500 steps: (a) output of the prediction-correction algorithm; the yellow part is the training data, and great is the prediction data in (b)

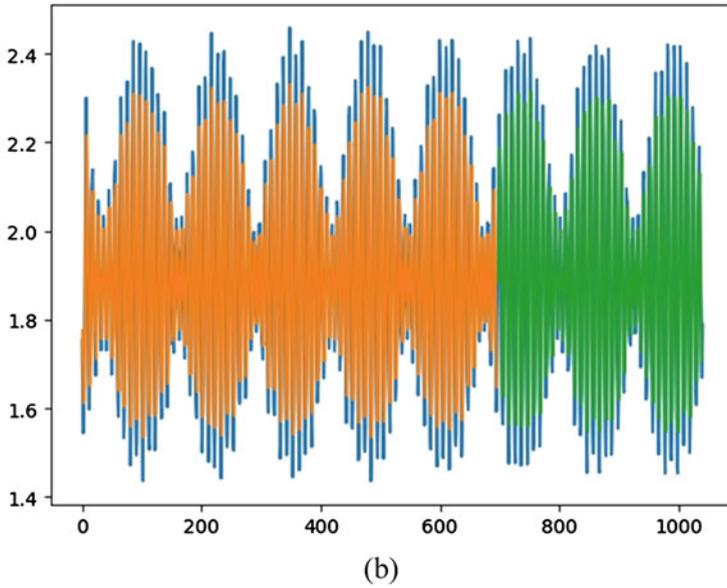
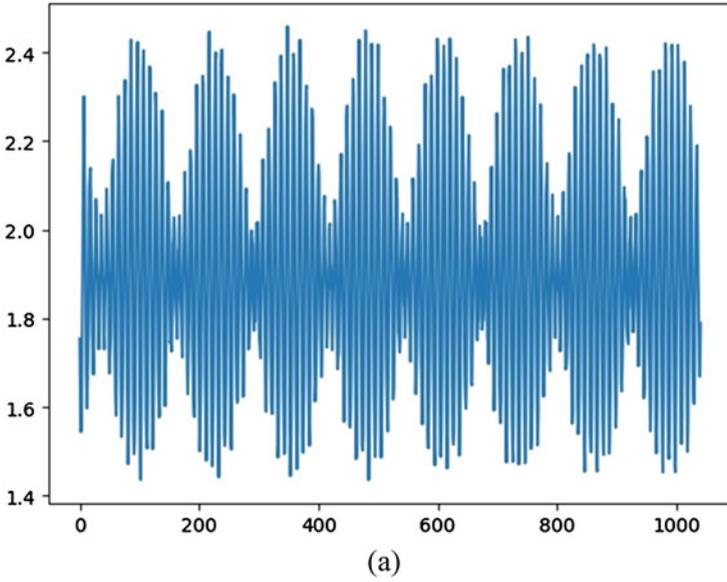


Fig. 11 The Hessian of CO₂ system with 5000 steps: (a) output of the prediction-correction algorithm; the yellow part is the training data, and great is the prediction data in (b)

Table 4 Relative error between Hessian prediction and computational value (500 steps)

Computational data (D1)	Predicted data D2	$ (D1-D2) / D1 $ (%)
2.1458862	2.0591743	4%
2.126515	2.0859475	1.9%
1.9289516	2.0703099	7.3%
1.8062238	1.9091784	5.7%
1.751231	1.8097774	3.3%

Acknowledgments This work was supported by Dr. Hase research group and cluster Chemdynm at Texas Tech University, the Industrial Internet Innovation and Development Project of China: Digital twin system for automobile welding and casting production lines and its application demonstration (TC9084DY).

References

1. D.L. Bunker, D. L. Bunker, Classical Trajectory Methods. *Comput. Phys.* **10**, 287–324 (1971)
2. J. Millam, V. Bakken, W. Chen, W.L. Hase, Ab initio classical trajectories on the Born–Oppenheimer surface: Hessian-based integrators using fifth-order polynomial and rational function fits. *J. Chem. Phys.* **111**, 3800–3805 (1999)
3. N. Sathyamurthy, Computational fitting of AB initio potential energy surfaces. *Comput. Phys. Rep.* **3**, 1–69 (1985)
4. H.-M. Keller, H. Floethmann, A.J. Dobbyn, R. Schinke, H.-J. Werner, C. Bauer, P. Rosmus, The unimolecular dissociation of HCO. II. Comparison of calculated resonance energies and widths with high-resolution spectroscopic data. *J. Chem. Phys.* **105**, 4983–5004 (1996)
5. X. Zhang, S. Zou, L.B. Harding, J.M. Bowman, A global ab initio potential energy surface for formaldehyde. *J. Phys. Chem.* **108**, 8980–8986 (2004)
6. H. Wu et al., Higher-accuracy schemes for approximating the hessian from electronic structure calculations in chemical dynamics simulations. *J. Chem. Phys.* **133**, 074101 (2010)
7. K.R. Müller, G. Rätsch, S. Sonnenburg, S. Mika, M. Grimm, N. Heinrich, Classifying ‘Drug-likeness’ with kernel-based learning methods. *J. Chem. Inf. Model.* **45**, 249–253 (2005)
8. A.P. Bartók, M.J. Gillan, F.R. Manby, G. Csányi, Machine-learning approach for one- and two-body corrections to density functional theory: Applications to molecular and condensed water. *Phys. Rev. B.* **88**, 054104 (2013)
9. NIH (2014), <https://ncats.nih.gov/news/releases/2015/tox21-challenge-2014-winners>.
10. M. Rupp, A. Tkatchenko, K.R. Müller, O.A. von Lilienfeld, Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.* **108**, 058301 (2012)
11. R. Ramakrishnan, P.O. Dral, M. Rupp, O.A. von Lilienfeld, Big data meets quantum chemistry approximations: the δ -machine learning approach. *J. Chem. Theor. Comput.* **11**, 2087 (2015)
12. L.U. Shao-fei, Z.E.N.G. Qian, W.U. Heng. A new power load forecasting model (SIndRNN): Independently recurrent neural network based on softmax kernel function, IEEE 21st international conference on high performance computing and communications (2019). <https://doi.org/10.1109/HPCC/SmartCity/DSS.2019.00320>.
13. Heng Wu, Shaofei Lu, Armando Lopez-Aeamburo, Jingke She. Temperature prediction based on long short term memory networks, CSCI’19 (2019)
14. V. Botu, R. Ramprasad, Adaptive machine learning framework to accelerate ab initio molecular dynamics. *Int. J. Quantum Chem.* **115**(16), 1074–1083 (2015)

15. E. Apra, T.L. Windus, T.P. Straatsma, et al., *NWChem, A Computational Chemistry Package for Parallel Computers, Version 5.0* (Pacific Northwest National Laboratory, Richland, Washington, 2007)
16. H. Wu, et al., A high accuracy computing reduction algorithm based on data reuse for direct dynamics simulations, CSCI (2016)
17. Heng Wu and Shaofei Lu, Evaluating the Accuracy of a Third Order Hessian-Based Predictor-Corrector Integrator, Europe Simulation Conference (2016)
18. H. Wu, S. Lu, et al., Evaluating the accuracy of hessian-based predictor-corrector integrators. *J. Cent. South Univ.* **24**(7), 1696–1702 (2017)

Evaluating the Effect of Compensators and Load Model on Performance of Renewable and Nonrenewable DGs



H. Shayeghi, H. A. Shayanfar, and M. Alilou

1 Introduction

The distribution system is the nearest part of the network to the consumers, so improving the efficiency of distribution network causes to increase the customers' satisfaction. Distributed generation and compensators are one of the useful devices for improving the technical, economic, and environmental indices of the distribution network. In the real system, various DGs are utilized to produce the demanded power so that nonrenewable units such as micro turbine are more trustworthy than renewable ones [1]. Compensators are used to increase the efficiency of DG and distribution system. Compensators can be divided into traditional and modern models. The main difference between the old and new technologies is using the power electronic devices in the modern compensators. Capacitor bank and D-FACT devices are the examples of traditional and modern compensators, respectively [2, 3].

In the last years, some studies are done on the issues of DGs and compensators [4–13]. Although various studies have been done on considered devices, in this article, the performance of capacitor bank and DSTATCOM is evaluated in the distribution network with different types of DG units. Simultaneous optimization of location and size of multi-DG and each type of compensator is done to evaluate the efficiency of devices in the distribution networks. For evaluating the devices in more

H. Shayeghi

Energy Management Research Centre, University of Mohaghegh Ardabili, Ardabil, Iran

H. A. Shayanfar (✉)

College of Technical & Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran

M. Alilou

Electrical Engineering Department, Urmia University, Urmia, Iran

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_18

235

realistic conditions, the load model of the system is considered as a combination of sensitive to voltage-frequency and various customers' daily load patterns. Technical, economic, and environmental indices of the system are objective functions that are optimized by a combination of multi-objective whale optimization algorithm (MOWOA) and analytical hierarchy process (AHP). The performance of devices is evaluated using the IEEE 69-bus distribution system.

2 Problem Description

2.1 Distributed Generation

Wind turbine The output power of wind turbine (WT) has direct communication with wind speed (V_w) and swept area (A_{wt}) of the turbine; however, the other parameters such as air density (ρ) and power coefficient (C_p) affect the power of WT. Therefore, the active power of WT can be evaluated by (1). WT consumes reactive power to produce active power. So in the load flow equations, it is modeled as a PQ bus model with variable reactive power. The reactive power consumed by a WT in a simple form is given in (2).

$$P_{WT} = \frac{1}{2} \rho A_{wt} V_w^3 C_p \quad (1)$$

$$Q_{WT} = - \left(0.5 + 0.04 P_{WT}^2 \right) \quad (2)$$

Photovoltaic The output power of each photovoltaic panel is related to the amount of solar irradiance (μ), the area (A_{pv}), and efficiency of the solar panel (β). Mathematically, the active power of PV can be calculated by (3). This DG produces only active power. So in the load flow analysis, PV is modeled as a P bus model.

$$P_{PV} = A_{pv} \beta \mu \quad (3)$$

Micro turbine Micro turbine (MT) units have the unique ability to simultaneously produce the electricity and heat. MT is capable of injecting both active and reactive power to the network. This kind of DG is modeled as a constant voltage bus model in load flow equations.

2.2 Compensators

Capacitor bank The capacitor bank is a common and cheap way to compensate the reactive power. A capacitor injects the required reactive power to improve the efficiency of the distribution system. So in the load flow equations, the capacitor can be modeled as a Q bus model.

DSTATCOM In this study, DSTATCOM is simulated as a PV bus model in load flow calculations. In other words, injected reactive power of DSTATCOM for voltage improvement of the connected bus can be expressed as (4) [9].

$$jQ_{DSTATCOM} = V_{jnew}(I_{DSTATCOM})^* \tag{4}$$

$$V_{jnew} = |V_{jnew}| < \alpha_{new} \tag{5}$$

$$I_{DSTATCOM} = |I_{DSTATCOM}| < ((\pi/2) + \alpha_{new}) \tag{6}$$

2.3 Load Model of Network

In this study, the nonlinear load model is considered to test the proposed method in more realistic conditions of operation in the distribution system. Load model of the system is considered as a combination of daily load pattern and sensitive load to voltage-frequency. Practical voltage-frequency-dependent load model can be mathematically expressed as the (7-8). The values of coefficients are shown in Table 1. Nominal and network frequencies are considered 1 and 0.98 Pu, respectively.

$$P_{li} = P_{l0i} \left(\frac{V_i}{V_b} \right)^{k_{pv}} [1 + k_{pf} (f - f_0)] \tag{7}$$

$$Q_{li} = Q_{l0i} \left(\frac{V_i}{V_b} \right)^{k_{qv}} [1 + k_{qf} (f - f_0)] \tag{8}$$

Table 1 Load types and values of dependence coefficients

Coefficients	Type of load			
	Constant	Residential	Industrial	Commercial
k_{pv}	0	1.7	0.1	0.6
k_{qv}	0	2.6	0.6	2.5
k_{pf}	0	1.0	2.6	1.5
k_{qf}	0	-1.7	1.6	-1.1

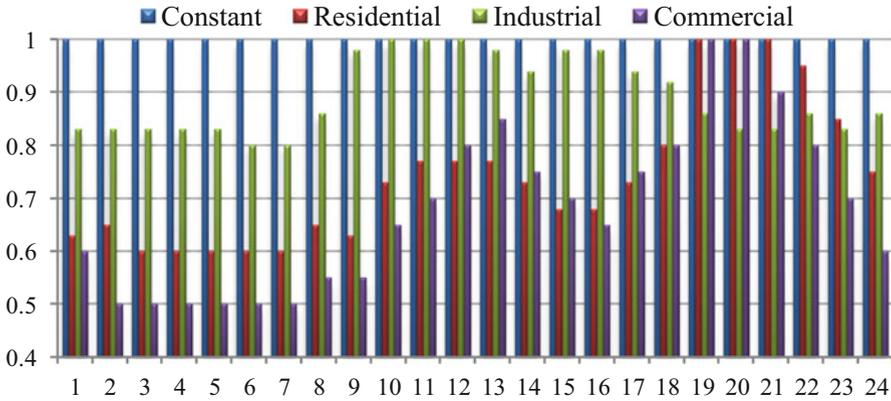


Fig. 1 Daily load demand of each type of load model (Pu)

In this paper, the load model of the network is also changed based on customers’ daily load patterns. Figure 1 shows the average hourly demand data in Pu for various types of customers in central Iran from 2011/2/20 to 2011/3/20. The magnitudes are normalized by the maximum of daily power demand [14].

2.4 Objective Functions

Technical, economic, and environmental indices of the distribution system are considered as objective functions of optimization. Mathematically, the main objective function is formulated as:

$$\text{objective function : } \min \{I_{\text{Tech}}, I_{\text{Env}}\}, \max \{I_{\text{Eco}}\} \tag{9}$$

2.5 Technical Index

The technical index is a combination of loss and voltage indices as the per unit format (10).

$$I_{\text{Tech}} = \frac{I_L}{I_{L0}} + \frac{I_V}{I_{V0}} \tag{10}$$

The Eqs. (11, 12, and 13) are used to calculate the loss index [1].

$$I_L = C_p \frac{P_1}{P_{10}} + C_q \frac{Q_1}{Q_{0l}} \tag{11}$$

$$\text{Active Loss} = \max_{h=1}^{24} \left\{ \sum_{i=1}^{N_{br}} R_i |I_{hi}|^2 \right\} \quad (12)$$

$$\text{Reactive Loss} = \max_{h=1}^{24} \left\{ \sum_{i=1}^{N_{br}} X_i |I_{hi}|^2 \right\} \quad (13)$$

Equations 14–16 are also used to calculate the voltage index. In this study, voltage stability is considered as voltage index. Voltage stability is the ability of the system to maintain the power and voltage in controllable condition [15].

$$I_V = V_S / V_{S0} \quad (14)$$

$$V_S = \max_{h=1}^{24} \{1 - \text{VoltageStability}_h\} \quad (15)$$

$$V_{S_{m_2}} = |V_{m_1}|^4 - \left\{ 4[P_{m_2} X_i - Q_{m_2} R_i]^2 \right\} - \left\{ 4|V_{m_1}|^2 [P_{m_2} R_i + Q_{m_2} X_i] \right\} \quad (16)$$

2.6 Economic Index

The economic index is defined according to (17).

$$I_{Eco} = \text{Profit} / \text{Initialprofit} \quad (17)$$

$$\text{Profit} = \text{Re} - \text{Co} \quad (18)$$

$$\text{Initialprofit} = \text{Re}_0 - \text{Co}_0 \quad (19)$$

Here, [16].

$$\text{Re} = \text{Re}_0 = \sum_{i=1}^{n_b} P_{li} \times C_{MR} \times 24 \quad (20)$$

$$\text{Co}_0 = \left[\sum_{i=1}^{n_b} P_{li} \times C_A \times 24 \right] + \left[\sum_{h=1}^{24} P_{lh} \times C_A \right] + \left[\sum_{h=1}^{24} Q_{lh} \times C_R \right] \quad (21)$$

$$\begin{aligned}
Co = & \left[\left(\sum_{i=1}^{n_b} P_{li} - \sum_{i=\text{tech}} \sum_{j=1}^{n_{DG}} P_{DGij} \right) \times C_A \times 24 \right] + \left[\sum_{h=1}^{24} P_{lh} \times C_A \right] \\
& + \left[\sum_{h=1}^{24} Q_{lh} \times C_R \right] + \left[\sum_{i=\text{tech}} \sum_{j=1}^{n_{DG}} P_{DGij} C_{DG_i} \right] + \left[\sum_{i=1}^{n_{co}} Q_{Co_i} \times C_{Co_i} \right] \quad (22)
\end{aligned}$$

2.7 Environmental Index

The environmental index is about the amount of produced pollutant gas from DG units during the day. The main pollutant gases are CO, CO₂, SO₂, NO_x, and PM₁₀. The environmental index can be calculated by (23) [16]. In this equation, TPG_{*i*} is the total amount of produced pollutant gases by DG units, and TPG_{Max} is the maximum amount of environmental pollution. Equation 24 is utilized for calculating the amount of produced pollutant gases.

$$I_{Env} = TPG_i / TPG_{Max} \quad (23)$$

$$TPG = \sum_{i=\text{tech}} \sum_{j=1}^{NDG_{\text{tech}}} \sum_{h=1}^{24} \sum_{g=1}^{n_g} P_{DG_{ijh}} \times WE_g \times EDG_{ig} \quad (24)$$

2.8 Optimization Method

For calculating the impact of DGs and compensators on the efficiency of distribution system, firstly, simultaneous optimization of location and size of multi-DG and each type of compensator is done by combination of MOWOA and AHP method. Then, the performance of network is evaluated using the results of objective functions of distribution system with and without considered devices.

Intelligent algorithms are typically inspired in their performance from nature; the whale optimization algorithm is inspired from the hunting behavior of whales. In the WOA, particles are updated based on “shrinking encircling mechanism” and “spiral updating position” during the optimization [17]. It is worth mentioning here that the used method in reference [18] has been used for multi-objective optimization. After obtaining the Pareto optimal set solution by MOWOA, the AHP method is utilized based on the importance of various indices of the distribution system to find the best result of the optimization [1].

2.9 Numerical Results

In this section, the impact of multi-DG, capacitor bank, and DSTATCOM is evaluated using the IEEE 69-bus radial distribution system. The maximum capacity of DG units is considered as 2.5 MW. The economic information is tabulated

in Table 2. The yearly fixed cost of capacitor banks is available in Ref [1]. The environmental information of DG units is also given in Table 3. The hourly variations of the produced energy of WT and PV are shown in Fig. 2. Now, the proposed optimization algorithm was applied to the 69-bus test system to determine the optimal location and size of devices in different combinations.

Table 4 shows the best location and size of multi-DG and compensators in different combinations of devices. The indices of 69-bus distribution system before and after operating of devices are presented in Table 5.

Loss index that has been consisted of active and reactive losses improves about 46–86 percent after operating of devices. The amount of improvement of loss index

Table 2 The economic information of network and devices

	Unit	Value		Unit	Value
C_A	\$/MWh	13.087	C_{PV}	\$/MWh	5.168
C_R	\$/MVarh	4.153	C_{MT}	\$/MWh	0.914
C_{WT}	\$/MWh	2.146	$C_{DSTATCOM}$	\$/Mvarh	6.849
Market price					
Load level	Period		C_{MP} (\$/MWh)		
Light	(23 < h < 7)		35		
Medium	(7 < h < 19)		49		
Peak	(19 < h < 23)		70		

Table 3 The environmental information of sources

	Pollution gases rate (Kg/Kwh)				
	CO ₂	SO ₂	NO _x	CO	PM ₁₀
MT	0.72	0.002	0.091	0.247	0.018
PV	0	0	0	0	0
WT	0	0	0	0	0
Grid	0.85	2.14	9.723	6.043	0.87

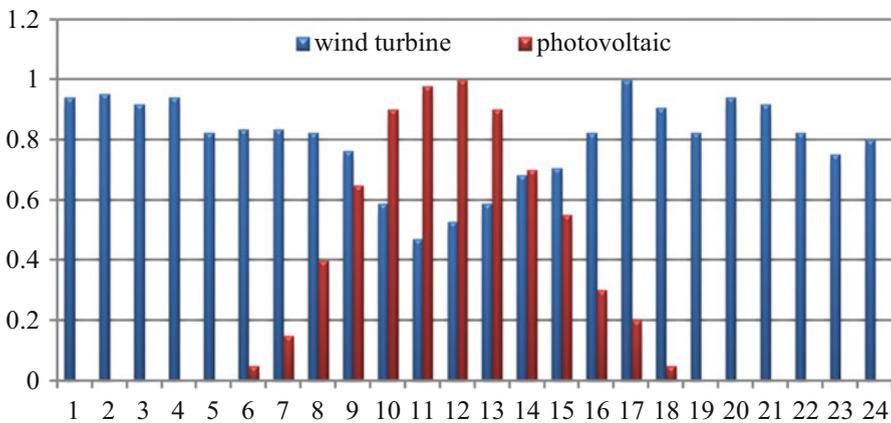


Fig. 2 The hourly variations of produced power of WT and PV

Table 4 The optimal result of devices

No. Test	Type of devices	DG		Compensator	
		Position (No. bus)	Capacity (MW)	Position (No. bus)	Capacity (Mvar)
1	PV and Ca	43	2.3272	42	0.9
2	WT and Ca	42	2.4780	42	1.2
3	MT and Ca	44	1.4085	33	1.2
4	PV and Ds	43	2.2524	45	1.2579
5	WT and Ds	42	2.2069	43	2.1975
6	MT and Ds	43	1.9660	26	1.2447

Ca capacitor, Ds DSTATCOM

in different tests is shown in Fig. 3. Base on this index, the combinations that have DSTATCOM have better performance than compounds of DG and capacitor bank.

Table 5 The indices of 69-bus distribution system during the optimization

No. Test	Load model	I_{Tech}			I_{Eco}	I_{Env}
		P_l (MW)	Q_l (Mvar)	V_s (Pu)	Profit (\$)	TPG (Mg)
Initial	Constant	0.2249	0.1021	0.6833	3116.22	1790.78
	Industrial	0.1947	0.0887	0.7024	2798.68	1596.78
	Commercial	0.1809	0.0831	0.7114	2269.58	1219.97
	Residential	0.1664	0.0770	0.7244	2458.76	1325.93
1	Constant	0.0363	0.0198	0.8984	3608.42	737.00
	Industrial	0.0391	0.0191	0.9062	3265.37	543.00
	Commercial	0.0633	0.0265	0.9028	2708.84	166.19
	Residential	0.0486	0.0211	0.9029	2903.35	272.15
2	Constant	0.0422	0.0217	0.9035	3831.73	623.58
	Industrial	0.0512	0.0228	0.9111	3487.24	429.58
	Commercial	0.0813	0.0336	0.9074	2929.26	52.77
	Residential	0.0633	0.0267	0.9071	3124.61	158.73
3	Constant	0.0268	0.0174	0.8819	3596.06	1163.78
	Industrial	0.0291	0.0174	0.8886	3257.28	969.78
	Commercial	0.0529	0.0274	0.8862	2705.06	592.97
	Residential	0.0386	0.0213	0.8864	2897.66	698.93
4	Constant	0.0371	0.0203	0.9028	3404.67	729.84
	Industrial	0.0468	0.0218	0.9105	3060.12	535.84
	Commercial	0.0772	0.0327	0.9068	2502.30	159.03
	Residential	0.0590	0.0258	0.9066	2697.63	264.98
5	Constant	0.0436	0.0218	0.9097	3399.44	751.27
	Industrial	0.0630	0.0274	0.9172	3052.91	557.27
	Commercial	0.1017	0.0419	0.9133	2493.54	180.46
	Residential	0.0785	0.0328	0.9127	2690.00	286.42
6	Constant	0.0321	0.0186	0.8967	3556.65	915.61
	Industrial	0.0269	0.0159	0.9045	3213.72	721.61
	Commercial	0.0284	0.0168	0.9014	2657.37	344.80
	Residential	0.0283	0.0168	0.9016	2851.77	450.75

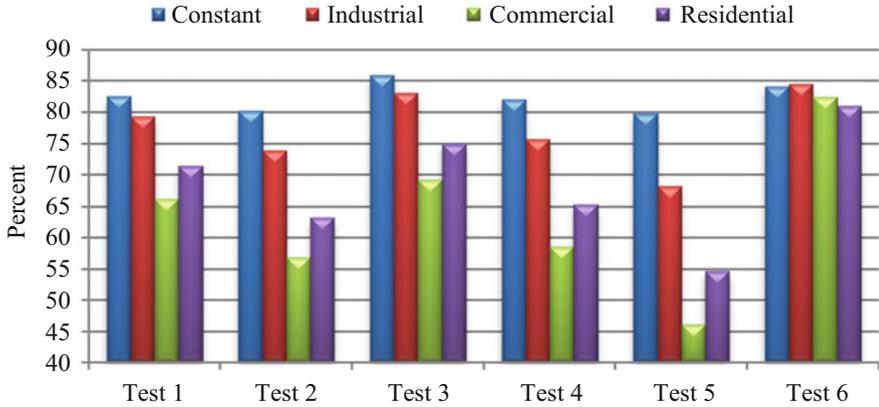


Fig. 3 The improvement of loss index in different experiments

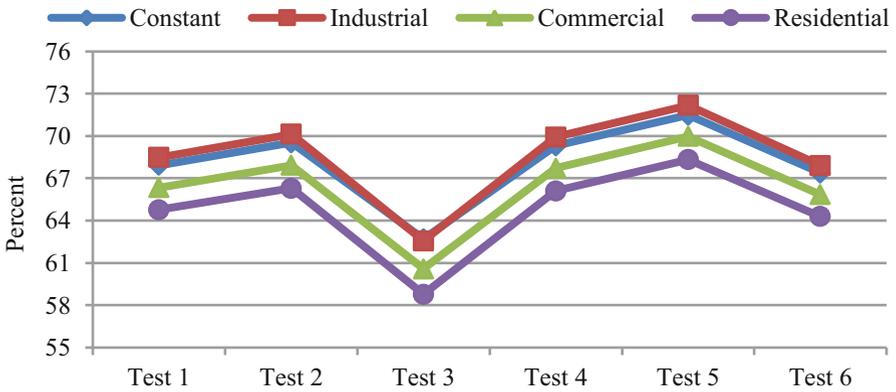


Fig. 4 The improvement of voltage stability in different experiments

According to Fig. 3, the located devices have proper performance in constant, industrial, residential, and commercial load model, respectively. The improvement of voltage index is about 59–72 percent in different experiments. Figure 4 shows the amount of increasing the voltage stability after placement of devices. According to this figure, capacitor bank has better performance than DSTATCOM so that the combinations that have capacitor increase the stability of voltage more than others. Based on voltage index, against loss index, the located devices have proper performance in industrial, constant, commercial, and residential load model, respectively.

Figure 5 shows the amount of economic index before and after operating the devices in different experiments. The profitability of distribution systems increases about 300–700 \$ in various combination of devices in each day. Of course, the

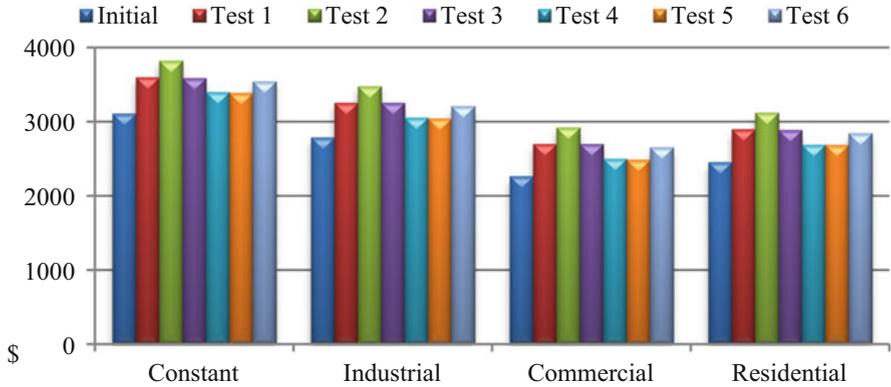


Fig. 5 The amount of economic index during optimization in each day

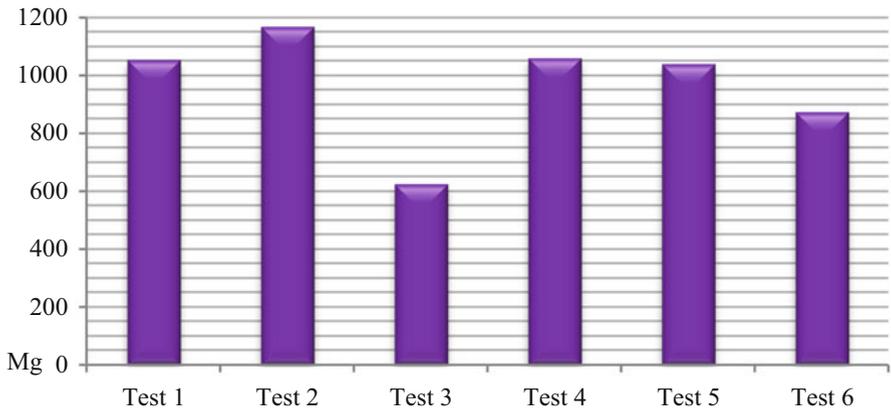


Fig. 6 The improvement of environmental index in each day

compounds that have capacitor have more profit than combinations of DGs and DSTATCOM.

The combinations of multi-DG and compensators are also environmentally friendly so that the amount of environmental index is improved approximately 50% after operating of devices. Of course, renewable DGs have more effect on this index than nonrenewable ones. The improvement of the environmental index in different experiments is shown in Fig. 6.

3 Conclusion

The results show that the combinations of multi-DG and compensators have practical performance in the various load models and improve the considered indices of the distribution system. Based on the DG types, it can be said that all kinds of DG technologies have useful effect on the efficiency of the distribution system; of course, the produced power of nonrenewable units is more stable than renewable DGs, while based on environmental aspects, renewable technologies are more useful than nonrenewable ones. Each type of compensators has proper performance beside of DG units based on special indices. Totally, the combinations of multi-DG and capacitor bank or DSTATCOM improve the efficiency of distribution system during the day in different load models.

References

1. H. Shayeghi, M. Alilou, Application of multi objective HFAPSO algorithm for simultaneous placement of DG, capacitor and protective device in radial distribution network. *J. Oper. Autom. Power Eng.* **3**, 131–146 (2015)
2. K.R. Padiyar, *FACTS Controllers in Power Transmission and Distribution* (New Age International Limited Publishers, New Delhi, 2007)
3. M. Aman, G. Jasmon, A. Bakar, H. Mokhlis, Optimal shunt capacitor placement in distribution system-a review and comparative study. *Renew. Sust. Energy. Rev.* **30**, 429–439 (2014)
4. S. Kansal, V. Kumar, B. Tyagi, Optimal placement of different type of DG sources in distribution networks. *Electr. Power Energy Syst.* **23**, 752–760 (2013)
5. B. Poornazaryan, P. Karimyan, G.B. Gharehpetian, M. Abedi, Optimal allocation and sizing of DG units considering voltage stability, losses and load variations. *Electr. Power Energy Syst.* **79**, 42–52 (2016)
6. H. Karimi, R. Dashti, Comprehensive Framework for Capacitor Placement in Distribution Networks from the Perspective of Distribution System Management in a Restructured Environment. *Electr. Power Energy Syst.* **82**, 11–18 (2016)
7. J. Gholinezhad, R. Noroozian, A. Bagheri, Optimal capacitor allocation in radial distribution networks for annual costs minimization using hybrid PSO and sequential power loss index based method. *J. Oper. Autom. Power Eng.* **5**, 117–130 (2017)
8. T. Yuvaraja, K.R. Devabalajia, K. Ravia, Optimal placement and sizing of DSTATCOM using harmony search algorithm. *Energy Procedia* **79**, 759–765 (2015)
9. A. Taher, A. Afsari, Optimal location and sizing of DSTATCOM in distribution systems by immune algorithm. *Electr. Power Energy Syst.* **60**, 34–44 (2014)
10. A. Khodabakhshian, M. Andishgar, Simultaneous placement and sizing of DGs and shunt capacitors in distribution systems by using IMDE algorithm. *Electr. Power Energy Syst.* **82**, 599–607 (2016)
11. H.A. Shayanfar, H. Shayeghi, M. Alilou, “*Multi-Objective Allocation of DG Simultaneous with Capacitor and Protective Device Including Load Model*”, the 19th International Conference on Artificial Intelligence (ICAI 2017) (Las Vegas, Nevada, USA, July 2017)
12. H. Tolabi, M. Ali, M. Rizwan, Simultaneous reconfiguration, optimal placement of DSTATCOM and photovoltaic Array in a distribution system based on Fuzzy-ACO approach. *IEEE Trans. Sustain. Energy* **6**, 210–218 (2015)
13. S. Devi, M. Geethanjali, Optimal location and sizing determination of distributed generation and DSTATCOM using particle swarm optimization algorithm. *Electr. Power Energy Syst.* **62**, 562–570 (2014)

14. R. Ebrahimi, M. Ehsan, H. Nouri, A profit-centric strategy for distributed generation planning considering time varying voltage dependent load demand. *Electr. Power Energy Syst.* **44**, 168–178 (2013)
15. S.M. Sajjadi, M.R. Haghifam, J. Salehi, Simultaneous placement of distributed generation and capacitors in distribution networks considering voltage stability index. *Electr. Power Energy Syst.* **46**, 366–375 (2013)
16. A. Zangeneh, S. Jadid, A. Rahimi-Kian, A fuzzy environmental-technical-economic model for distributed generation planning. *Energy* **36**, 3437–3445 (2011)
17. S. Mirjalili, A. Lewis, The whale optimization algorithm. *Adv. Eng. Softw.* **95**, 51–67 (2016)
18. C.A. Coello Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization. *IEEE Trans. Evol. Comput.* **8**, 256–279 (2004)

The Caloric Curve of Polymers from the Adaptive Tempering Monte Carlo Method



Greg Helmick, Yoseph Abere, and Estela Blaisten-Barojas

1 Introduction

Polypyrrole (PPy) is a prototypical conducting polymer formed by heterocyclic aromatic monomers that are able to change its volume by 30% depending upon its pristine (neutral) or doped (oxidized) chemical phases. Current computational approaches have had difficulty calculating properties that agree well with experimental results. In the past, our group developed a coarse-grained force field for pristine 12 PPy (12 monomers per chain) in which the pyrrole monomers are modeled by a planar 5-member ring with a permanent dipole moment pointing from the monomer center of mass to the nitrogen atom [1]. We have extended that force field for modeling the oxidized phase of PPy, which entails including dopants and taking into account a number of polymer-dopant interactions. The new force field has intra-chain terms similar to the previous model and new interchain interaction components, E_{int} , including the interaction between monomers and dopants, between dopants and dopants, and between monomers in a 12-PPy chain with monomers in a different chain [2]. In total, the full model potential has 15 parameters. Polymer chains with 12 monomers per oligomer (12 PPy) are consistent with the synthetic rendering of this doped polymeric material [3]. Dopants trapped in the polymer matrix result in the oxidized phase of PPy by inducing a charge transfer between the polymer and the dopants. In our case, the dopants acquire negative charge, and the polymer chains lose electrons becoming positively charged. Polymer matrix volume changes are dependent on the size and type of dopants used.

G. Helmick · Y. Abere · E. Blaisten-Barojas (✉)
Center for Simulation and Modeling and Department of Computational and Data Sciences,
George Mason University, Fairfax, VA, USA
e-mail: ghelmick@masonlive.gmu.edu; yabere@gmu.edu; blaisten@gmu.edu

© Springer Nature Switzerland AG 2021
H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_19

In this work, we explore with a novel numerical simulation implementation the ability to reproduce the volume expansion of PPy when oxidized. In Sect. 2, we describe the custom parallelization implementation for the calculation of the thermodynamic and mechanical properties of PPy using the newly developed force field. Section 3 provides the calculated PPy properties obtained from the caloric curve for a case system containing chlorine as the atomic dopants. Section 4 concludes this paper.

2 Methods

Although being a molecular component, each 12-PPy monomer will be referred to as a *particle*. Dopants are atoms and will also be referred to as particles. Systems ranging in size from 1024 to 19200 particles were explored for calibrating the size scaling effect on the computational implementation. Simulations were performed with our custom-developed horizontally scalable GPU accelerated Metropolis Monte Carlo (MMC) implementation that utilizes the adaptive tempering Monte Carlo (ATMC) method [4] for the sampling exploration of configuration space as the parallelization strategy for intermolecular energy calculations. The ATMC visits different NPT Gibbs ensembles along with the determination of the caloric curve (enthalpy as a function of temperature). Indeed, the ATMC algorithm drives the evolution of the simulation by selecting which temperature should be explored along the caloric curve. Our numerical implementation supports four simulation modes: NVT-MMC (canonical ensemble), NPT-MMC (isobaric-isothermal ensemble), NVT-ATMC, and NPT-ATMC.

Parallelization of atomistic simulations typically falls into one of the three categories: embarrassingly parallel or replica exchange, domain decomposition, and farm or energy decomposition. The embarrassingly parallel or replica exchange method maintains several independent replicas of the full system configuration, each of them sets at a different temperature. A random walk is achieved by periodic exchanges of configurations at nearby temperatures. The domain decomposition method utilizes a spatial decomposition approach that allows for simultaneous updates of all particles. The latter is accomplished by moving particles that are outside each other's cutoff radius. The farm or energy decomposition approach achieves parallelization by splitting the particles into groupings and then calculating partial energy values for each group. The latter are combined to determine the total system energy. The parallelization approach selected for this research is a variant of the farm or energy decomposition method and contains components of previous MMC parallel implementations [5].

3 Results

3.1 Parallelization Protocol for the PPy Condensed Phase Analysis

The parallelization of our new potential model involves three levels of parallelism. The first level of parallelism is provided by the usage of Open MPI, which provides support for distributed memory systems such as computer clusters. The second tier of parallelization is provided by OpenMP, which is designed for shared memory architectures and is effective within one node of a computer cluster. The third tier of parallelization is provided by CUDA. This combination of parallelization technologies allows for an optimal utilization of current and future computational platforms. The combined approach of the three-level parallelization scheme is scalable to any sized system as specific care is taken in selecting the number of nodes and processes so that internodal communication is minimized. The three-level parallelization strategy is accomplished by partitioning the internal summations of the model potential terms into sum segments, with each sum segment assigned to a computational node. Each computational node evaluates partial one-particle energies associated with the sum segment it was assigned. Each node utilizes a combination of OpenMP and CUDA to further speed calculations.

The parallelization performance using the customized energy decomposition approach is schematically shown in Fig. 1, illustrating the processing time of a single MMC passage over all the particles when adopting different loads to the OpenMP (threads), Open MPI (processors), or GPU. As evident from the plot, when the number of particles is below 10000 (Fig. 1 inset), the selection of how to combine processor and thread components is important because the CPU time may be doubled if care is not taken. As the system is scaled up in size to many more particles, the balance processor/thread is still important but not as damaging. Also evident from Fig. 1 is that the processing time performance is the best when using the GPU, if the latter is available. We emphasize that these performance metrics

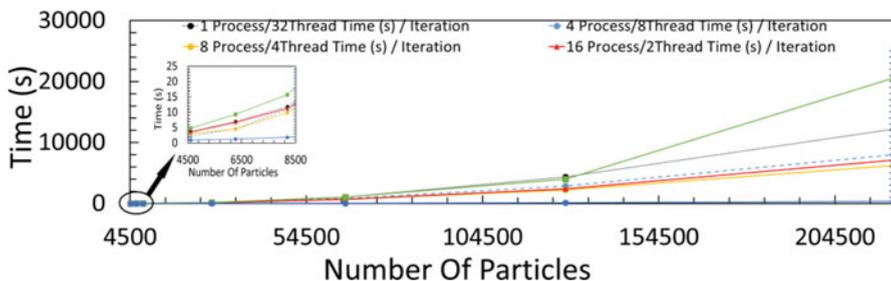


Fig. 1 Performance metrics for the parallelization scheme processing time of one single MMC iteration involving one passage over all the particles in the PPy system. The inset provides a better view of the processing time differences for PPy systems with less than 10000 particles

involve only one MMC step. In a regular simulation for obtaining the physical properties of the PPy system, several millions of these steps are needed.

The overall analysis of the PPy system within a single node with GPU involves a series of steps that we summarize in the flowchart depicted in Fig. 2. The control node initializes the system of particles and then shares the system with worker nodes. If the system is run in one node, the implementation calls one worker (Fig. 2); otherwise, several workers may be called as illustrated in Fig. 3. Once the PPy system is constructed, the spatial coordinates for the monomer and dopants will be distributed to all processing nodes. One node is designated as the control node and will perform all the reduction operations for the partial energies and evaluated the Monte Carlo acceptance step criteria. Initially, the control node is also given a sequence of random numbers and a set of random moves equal to the number of particles in the system. Next, the sum segmentation of model potential terms is determined based on the total number of monomers and dopants in the system and the number of computational nodes available. Next, the random translation/rotation MMC move of each particle is calculated on the control node, and the coordinates of the moved particle are shared with the other computational nodes where the new potential energy is calculated by segments. The computing nodes share their calculated partial energies with the control node where the sum of the partial energies is executed. The workflow enters next into the MMC main loop by accepting or rejecting the monomer or dopant move (translation/rotation). It is the control node that evaluates what action should occur next. If the particle move is accepted, the simulation evolves to the next step of execution. If the particle move is rejected, a reset message is broadcast to all compute nodes to reset values. Although the acceptance/rejection of the particle move is done in the control node, the potential energy associated with the moved particle is distributed between the computing nodes. To accept a move, the control node checks that the new potential energy be lower than the current or if the potential energy is larger, then the decision is done by checking that the change in energy gives a probability $e^{-(E_{new}-E_{old})/k_B T}$ larger or smaller than a random number in the range of 0–1, where k_B is Boltzmann's constant and T is the set system temperature. The particle motion step size is dynamically updated to ensure a 40% to 60% acceptance rate on the particle movement. For efficiency purposes, the control node evaluates the PPy intra-chain terms of the force field. The intra-chain potential energy is parallelized on the control node using OpenMP; this does place more computation on the control node but minimizes internodal communication overhead from Open MPI. The 12-PPy oligomers explored in this research contain 12 monomers only. Thus, the parallelization of the intra-oligomer potential energy represents a minimal gain that, however, could become important if longer polymer chains are considered in the future.

For a multi-compute node configuration, after the initialization step, the work distribution step described previously would occur. Figure 3 illustrates the work distribution for a multi-compute node configuration and a hypothetical number of 60,000 particles. In the multi-compute node configuration, each node is responsible for the evaluation of only part of the E_{int} . In the instance where a GPU is available

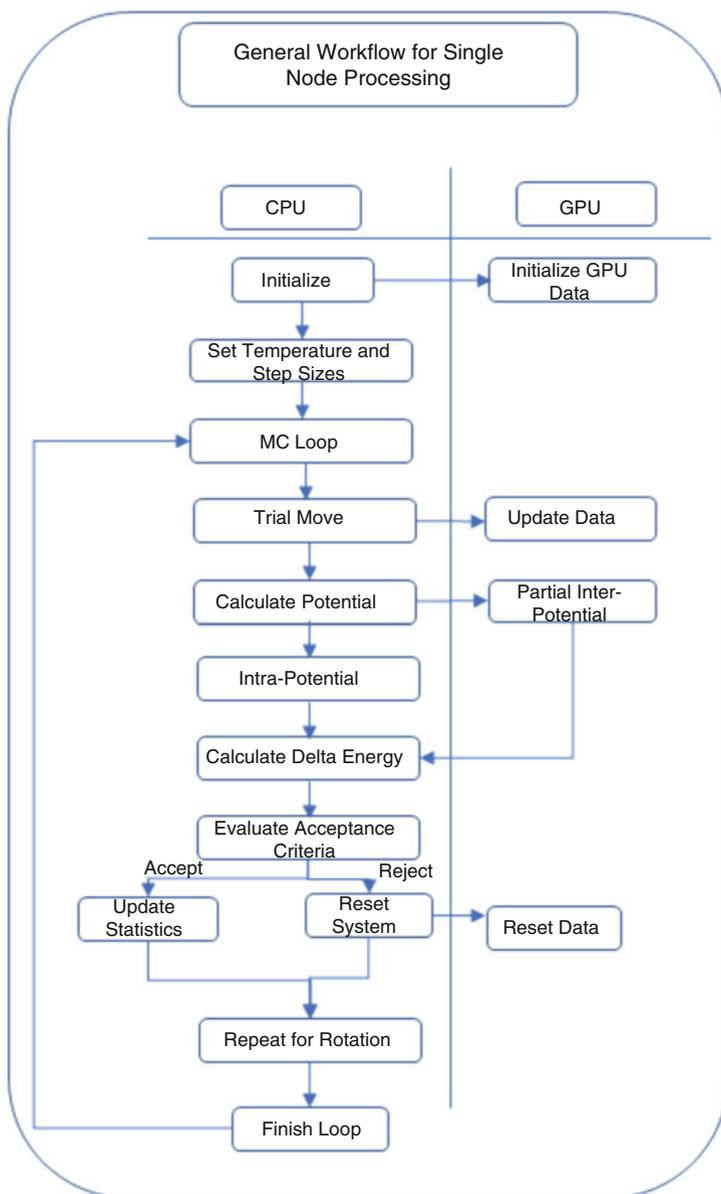


Fig. 2 General process workflow for the simulation utilized in this research

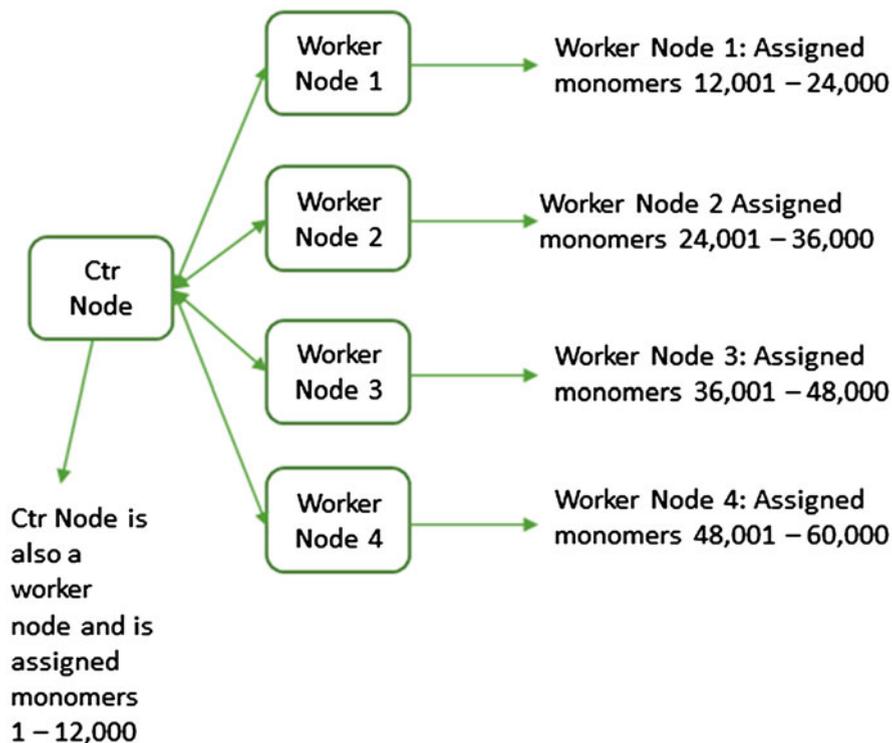


Fig. 3 An example work decomposition using four nodes for a system consisting of 60,000 particles. The control node will be responsible for the delegation and management of all work assignments

on each of the compute nodes, each compute node will off-load the E_{int} evaluation to the GPU.

3.2 PPy Thermodynamic Properties

In this section, we focus primarily on providing the ATMC results for a mid-size oxidized PPy system consisting of 8192 particles. Figure 4 shows the calculated caloric curve for the PPy condensed system. Equilibrium density at ambient temperature is in the range of 1.20 g/cm³ to 1.22 g/cm³, which is close to the experimentally determined range of 1.30 to 1.46 g/cm³ [3]; bulk modulus was determined to be in the range of 67–120 MPa, which is within the range of published experimental work of 100–350 MPa [6]; and vector and orientational order parameters indicate that our model reproduces the planar stacking of the polymer chains and specific heat value of 855.93 J kg⁻¹K⁻¹ at 300 K that are within the experimental range of values of

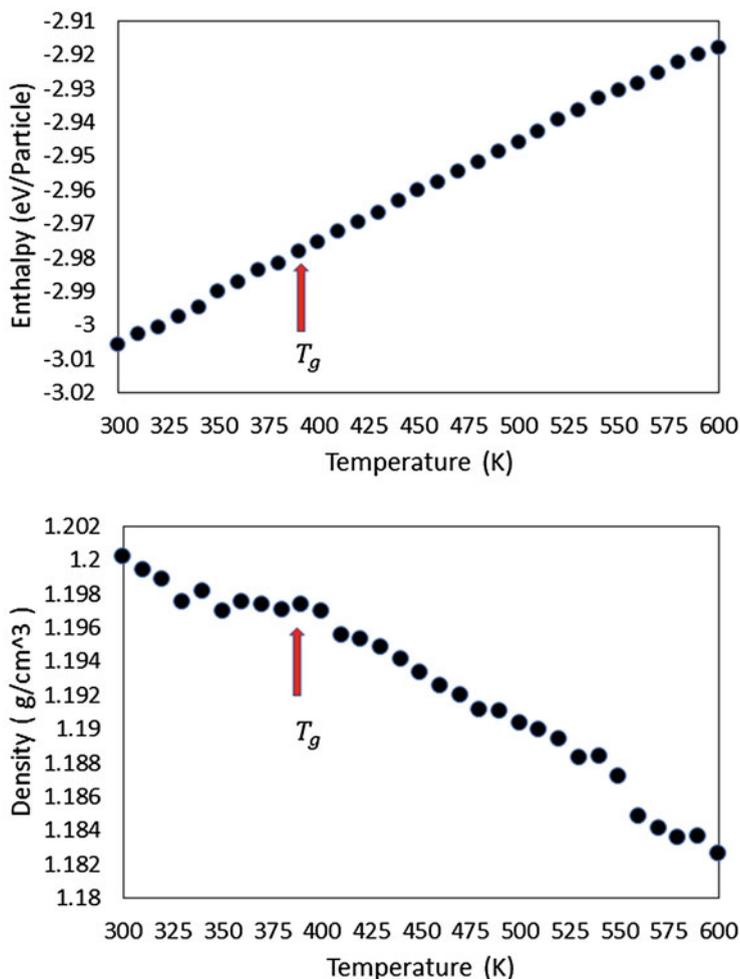


Fig. 4 Caloric curve results from the ATMC simulation beginning at 1000 K and ending at 300 K at 1 atm pressure for a particle system of 6144 pyrrole monomers and 2048 chlorine atom dopants

800–1400 J kg⁻¹K⁻¹ [7]. The glass transition temperature was determined to be 400.8 K, which is in very good agreement with the experimentally determined value of 394 K [8].

Concerning the structure of the 12-PPy chains, remarkably, they remain quite stiff along these extensive simulations maintaining a quasi-planar geometry even at high temperatures. The radius of gyration, end-to-end distance, and orientational order parameter change at most by 5% from the initial fully planar conformation.

4 Conclusion

The new force field accurately produces results consistent with values obtained experimentally for oxidized PPy systems. Our model reproduces the stacking of planar PPy chains observed in the experimental microstructure. We predict a glassy structure below the glass transition of 400.8 K. Other calculated properties of this glassy condensed system include the heat capacity, thermal expansion, and structural order parameters, which agree well with experimental and observation data.

The ATMC method provided an efficient and fast method for thoroughly exploring the configuration space of the PPy system. The ATMC provided significant computational time savings even though it visited a larger number of ensembles. The customized energy decomposition approach combined with a hybrid OpenMP-Open MPI multiprocessing strategy allows for the horizontal scalability to accommodate system configurations of a wide range of sizes, accomplished by minimizing data transfers between computational resources. Currently, we are applying this implementation to other complex systems.

Acknowledgments Partial support from the Commonwealth of Virginia 4-VA grant 331050 is acknowledged. Computations were done in the HPC platform of the Office for Research Computing, George Mason University.

References

1. Y. Dai, E. Blaisten-Barojas, Monte Carlo study of oligopyrroles in condensed phases. *J. Chem. Phys.* **133**, 034905 (2010)
2. G. Helmick, Y. Abere, E. Blaisten-Barojas, New force field for oxidized polypyrrole, to be submitted (2020)
3. J.M. Fonner, C.E. Schmidt, P. Ren, A combined Molecular Dynamics and experimental study of doped polypyrrole. *Polymer* **51**, 4985–4993 (2010)
4. X. Dong, E. Blaisten-Barojas, Adaptive tempering Monte Carlo method. *J. Comput. Theor. Nanosci.* **3**, 118–127 (2006)
5. C. Hall, W. Ji, E. Blaisten-Barojas, The Metropolis Monte Carlo method with CUDA enabled graphic processing units. *J. Comput. Phys.* **258**, 871–879 (2014)
6. R.Z. Pytel, E.L. Thomas, I.W. Hunter, In situ observation of dynamic elastic modulus in polypyrrole actuators. *Polymer* **49**, 2008–2013 (2008)
7. A. Rudajevová, M.P.J. Varga, S.J. Kopecka, Thermal properties of conducting polypyrrole nanotubes. *Acta Phys. Pol. A* **128**, 730–736 (2015)
8. P. Mavinakuli, S.W. Wei, A.B. Karki, S. Dhage, Z. Wang, D.P. Young, Z. Guo, Polypyrrole-silicon carbide nanocomposites with tunable electrical conductivity. *J. Phys. Chem. C* **114**, 3874–3882 (2010)

Part IV
Scientific Computing, Computational
Science, and Applications

A New Technique of Invariant Statistical Embedding and Averaging in Terms of Pivots for Improvement of Statistical Decisions Under Parametric Uncertainty



Nicholas A. Nechval, Gundars Berzinsh, and Konstantin N. Nechval

1 Introduction

In this chapter, we propose a novel technique of invariant statistical embedding and averaging in terms of pivotal quantities (ISE&APQ) to solve the problems of estimation, improvement, or optimization of statistical decisions under parameter uncertainty. The technique of ISE&APQ, the idea of which belongs to the authors, is based on the constructive use of the invariance principle in mathematical statistics and allows one to solve many problems of the theory of statistical inferences in a simple way. It allows one to yield operational, optimal information-processing rules and may be employed for finding efficient statistical decisions for problems such as multi-product newsboy problems with constraints, allocation problems of aircraft to routes under parametric uncertainty, airline seat inventory control problems for multi-leg flights, etc. The chapter is concerned with the implications of group theoretic structure for invariant performance criteria.

The aim of this chapter is to show how the technique of ISE&APQ may be employed in the particular case of optimization, estimation, or improvement of statistical decisions under parametric uncertainty. The technique used here is a special case of more general considerations applicable whenever the statistical problem is invariant under a group of transformations, which acts transitively on the parameter space [1–5].

N. A. Nechval (✉) · G. Berzinsh
BVEF Research Institute, University of Latvia, Riga, Latvia
e-mail: nechval@junik.lv

K. N. Nechval
Transport and Telecommunication Institute, Aviation Department, Riga, Latvia
e-mail: konstan@tsi.lv

2 Preliminaries

In the general formulation of decision theory, we observe a random sample $\mathbf{X} = (X_1, X_2, \dots, X_n)$ with a sequence of iid rv's with common distribution function $F_\theta(x)$ where the parameter θ (in general, vector) is unknown, $\theta \in \Theta$ (parameter space). A statistic $S = S(\mathbf{X})$ is sufficient for θ or for the family of distributions $\{F_\theta : \theta \in \Theta\}$ if and only if the conditional distribution of \mathbf{X} , given $S = s$, does not depend on θ . If we choose decision d from the set of all possible decisions \mathbf{D} , then we suffer a loss $L(\theta, d)$. A “decision rule” is a method of choosing d from \mathbf{D} after observing S , that is, a function $d(S) = d$. Our average loss (called risk) $E_\theta\{L(\theta, d(S))\}$ is a function of both θ and the decision rule $d(\cdot)$, called the risk function $R(\theta, d)$, and is the criterion by which rules are compared. Thus, the expected loss (gains are negative losses) is a primary consideration in evaluating decisions. We will now define the major quantities just introduced.

Definition 1 A general statistical decision problem is a triplet (Θ, \mathbf{D}, L) and a random sample \mathbf{X} . The random variable X from \mathbf{X} (called the data) has a distribution function $F_\theta(x)$ where θ is unknown but it is known that $\theta \in \Theta$. \mathbf{X} will denote the set of possible values of the random variable X . θ is called the state of nature, while the nonempty set Θ is called the parameter space. The nonempty set \mathbf{D} is called the decision space or action space. Finally, L is called the loss function, and to each $\theta \in \Theta$ and $d \in \mathbf{D}$, it assigns a real number $L(\theta, d)$.

Definition 2 For a statistical decision problem (Θ, \mathbf{D}, L) , \mathbf{X} , a (nonrandomized) decision rule is a function $d(\cdot)$ which to each S assigns a member d of \mathbf{D} : $d(S) = d$.

Definition 3 The risk function $R(\theta, d)$ of a decision rule $d = d(S)$ for a statistical decision problem (Θ, \mathbf{D}, L) , \mathbf{X} (the expected loss or average loss when θ is the state of nature and a decision is chosen by rule $d(\cdot)$) is $R(\theta, d) = E_\theta\{L(\theta, d)\}$.

This chapter is concerned with the implications of group theoretic structure for invariant loss functions. Our underlying structure consists of a class of probability models $(\mathbf{X}, \mathbf{A}, \mathbf{P})$, a one-one mapping ψ taking \mathbf{P} onto an index set Θ , a measurable space of actions (\mathbf{D}, \mathbf{B}) , and a real-valued loss function $L(\theta, d)$ defined on $\Theta \times \mathbf{D}$. We assume that a group G of one-one \mathbf{A} – measurable transformations acts on \mathbf{X} and that it leaves the class of models $(\mathbf{X}, \mathbf{A}, \mathbf{P})$ invariant. We further assume that homomorphic images \bar{G} and \tilde{G} of G act on Θ and \mathbf{D} , respectively. (\bar{G} may be induced on Θ through ψ as in [6]; and \tilde{G} may be induced on \mathbf{D} through L , see [7]). We shall say that L is invariant if for every $(\theta, d) \in \Theta \times \mathbf{D}$

$$L(\bar{g}\theta, \tilde{g}d) = L(\theta, d), \quad g \in G. \quad (1)$$

Let us assume that a loss function, $L(\theta, d)$, can be transformed as follows:

$$L(\theta, d) = L(\bar{g}_S\theta, \tilde{g}_Sd) = L^\#(V, \eta), \quad (2)$$

where $V = V(S, \theta)$ is a pivotal quantity whose distribution does not depend on unknown parameter θ , $\eta = \eta(S, d)$ is an invariant decision function, S is a sufficient statistic for θ (or a maximum likelihood estimator of θ). Then a risk function is given by

$$R(\theta, d) = E_{\theta} \{L(\theta, d)\} = E \left\{ L^{\#}(V, \eta) \right\}, \tag{3}$$

where the unknown parameter θ is eliminated from the problem. In this case, the statistical decision rule d which minimizes the risk described by (3) is given by

$$d = \eta^{*-1}(S, d), \tag{4}$$

where

$$\eta^* = \arg \min_{\eta} E \left\{ L^{\#}(V, \eta) \right\}. \tag{5}$$

Consider now, for example, the problem of estimating the location-scale parameter of a distribution belonging to a family generated by a continuous cumulative distribution function (CDF) F .

$$P = \left\{ P_{\theta} : F_{\theta}(x) = F \left(\frac{x - \mu}{\sigma} \right), x \in R, \theta \in \Theta \right\},$$

$$\Theta = \{(\mu, \sigma) : \mu, \sigma \in R, \sigma > 0\} = D. \tag{6}$$

The group G of location and scale changes leaves the class of models invariant. Since \bar{G} induced on Θ by $P_{\theta} \rightarrow \theta$ is uniquely transitive, we may consider invariant loss function of the form

$$L(\theta, d) = L \left(\frac{d_1 - \mu}{\sigma}, \frac{d_2}{\sigma} \right) \tag{7}$$

where $\theta = (\mu, \sigma)$ and $d = (d_1, d_2)$. Let us assume that there is the maximum likelihood estimator $\hat{\theta} = (\hat{\mu}, \hat{\sigma})$ of $\theta = (\mu, \sigma)$. Then a loss function, $L(\theta, d)$, can be transformed as follows:

$$\begin{aligned} L(\theta, d) &= L \left(\bar{g}_{\hat{\theta}} \theta, \tilde{g}_{\hat{\theta}} d \right) = L \left(\frac{d_1 - \hat{\mu} \hat{\sigma}}{\hat{\sigma}} + \frac{\hat{\mu} - \mu}{\sigma}, \frac{d_2 \hat{\sigma}}{\hat{\sigma} \sigma} \right) \\ &= L(\eta_1 V_2 + V_1, \eta_2 V_2) = L^{\#}(V, \eta), \end{aligned} \tag{8}$$

where

$$V = \left(V_1 = \frac{\widehat{\mu} - \mu}{\sigma}, V_2 = \frac{\widehat{\sigma}}{\sigma} \right), \quad (9)$$

V_1, V_2 are pivotal quantities,

$$\eta = \left(\eta_1 = \frac{d_1 - \widehat{\mu}}{\widehat{\sigma}}, \eta_2 = \frac{d_2}{\widehat{\sigma}} \right), \quad (10)$$

η_1, η_2 are invariant decision functions. Then a risk function is given by

$$R(\theta, d) = E_{\theta} \{L(\theta, d)\} = E \left\{ L^{\#}(V, \eta) \right\}, \quad (11)$$

where the unknown parametric vector $\theta = (\mu, \sigma)$ is eliminated from the problem. In this case, the statistical decision rules d_1, d_2 , which minimize the risk described by (11), are given by

$$d_1 = \eta_1^{*-1}(\widehat{\mu}, \widehat{\sigma}, d_1) = \widehat{\mu} + \eta_1^* \widehat{\sigma}, \quad (12)$$

$$d_2 = \eta_2^{*-1}(\widehat{\sigma}, d_2) = \eta_2^* \widehat{\sigma}, \quad (13)$$

where

$$(\eta_1^*, \eta_2^*) = \eta^* = \arg \min_{\eta} E \left\{ L^{\#}(V, \eta) \right\}. \quad (14)$$

3 Technique of Invariant Statistical Embedding and Averaging in Terms of Pivotal Quantities (ISE&APQ)

The technique of ISE&APQ includes the following three stages:

Stage 1 (Invariant statistical embedding via pivotal quantities) At this stage, an invariant embedding of a sample statistic in the decision criterion (performance index) is carried out to construct a *pivotal quantity* (or simply a *pivot*) in order to isolate the unknown parameter from the problem (*since the pivot's probability distribution does not depend on the unknown parameter*).

Stage 2 (Averaging via pivotal quantities) At this stage, the decision criterion is averaged over the pivots' probability distributions in order to eliminate the unknown parameters from the problem.

Stage 3 (Decision-making process) At this stage, when the unknown parameters have been eliminated from the decision criterion, it can be found an effective statistical decision rule.

4 Comparison of Statistical Decision Rules

In order to judge which statistical decision rule might be preferred for a given situation, a comparison based on some “closeness to the true value” criteria should be made. The following approach is commonly used.

Consider two estimators, say, d_1 and d_2 having risk functions $R(\theta, d_1)$ and $R(\theta, d_2)$, respectively. Then the relative efficiency of d_1 relative to d_2 is given by

$$\text{rel. eff.}_R \{ d_1, d_2 | \theta \} = R(\theta, d_2) / R(\theta, d_1). \tag{15}$$

When $\text{rel. eff.}_R \{ d_1, d_2 | \theta_0 \} < 1$ for some θ_0 , we say that d_2 is more efficient than d_1 at θ_0 . If $\text{rel. eff.}_R \{ d_1, d_2 | \theta \} \leq 1$ for all θ with a strict inequality for some θ_0 , then d_1 is inadmissible relative to d_2 .

5 Example 1 (Quantile Estimation)

Consider, for example, the problem of estimating a quantile q of an exponential distribution on the basis of a random sample X_1, \dots, X_n of size $n \geq 2$. The exponential distribution is often used for the length of life data. The exponential probability density function (PDF) is given by

$$f_\theta(x) = \frac{1}{\theta} \exp\left(-\frac{x}{\theta}\right), \quad x > 0, \quad \theta > 0. \tag{16}$$

The cumulative distribution function (CDF) is given by

$$F_\theta(x) = 1 - \exp\left(-\frac{x}{\theta}\right), \quad x > 0, \quad \theta > 0. \tag{17}$$

Quantile estimation, particularly for the exponential distribution, is important in reliability theory, life testing, and so on. Also, in statistical decision theory, it is of interest to find out if the best equivariant estimator or the maximum likelihood estimator (MLE) of quantile is admissible.

Thus, the problem is to estimate the p^{th} quantile $q = \vartheta p$ of the exponential distribution, where $0 < \vartheta = -\ln(1 - p)$; $0 < p < 1$. The loss function is taken as

$$L(\theta, d) = (F_\theta(d) - p)^2, \tag{18}$$

where d is the estimator (decision rule) for estimating the quantile q . We evaluate the performance of an estimator for quantile with the help of the risk function (decision criterion).

$$R(\theta, d) = E_\theta \{L(\theta, d)\}. \tag{19}$$

Assuming that the parameter θ is unknown, we find the maximum likelihood estimator (MLE) of θ given by

$$\widehat{\theta} = \sum_{i=1}^n X_i/n. \tag{20}$$

It is known that

$$\widehat{\theta} \sim \varphi_\theta(\widehat{\theta}) = \frac{n^n}{\Gamma(n)\theta^n} \widehat{\theta}^{n-1} \exp\left(-\frac{n\widehat{\theta}}{\theta}\right), \quad \widehat{\theta} > 0, \quad \theta > 0, \tag{21}$$

where

$$V = \widehat{\theta}/\theta \tag{22}$$

represents the pivotal quantity with the probability density function

$$\varphi(v) = \frac{n^n}{\Gamma(n)} v^{n-1} \exp(-nv), \quad v > 0. \tag{23}$$

To solve the above problem, the technique of invariant statistical embedding and averaging via pivotal quantities (ISE&APQ), proposed in this chapter, can be used.

Using the technique of ISE&APQ, we have the following:

Stage 1 *Invariant embedding of the MLE $\widehat{\theta}$ in the decision criterion to construct the pivotal quantity V :*

$$\begin{aligned} R(\theta, d) &= E_\theta \{L(\theta, d)\} = E_\theta \left\{ (F_\theta(d) - p)^2 \right\} \\ &= E_\theta \left\{ \left(1 - \exp\left(-\frac{d}{\theta}\right) - p \right)^2 \right\} = E_\theta \left\{ \left(1 - p - \exp\left(-\frac{d}{\theta}\right) \right)^2 \right\} \end{aligned}$$

$$\begin{aligned}
 &= E_{\theta} \left\{ (1 - p)^2 - 2(1 - p) \exp\left(-\frac{d}{\theta}\right) + \exp\left(-\frac{2d}{\theta}\right) \right\} \\
 &= E_{\theta} \left\{ (1 - p)^2 - 2(1 - p) \exp\left(-\frac{d \widehat{\theta}}{\widehat{\theta} \theta}\right) + \exp\left(-\frac{2d \widehat{\theta}}{\widehat{\theta} \theta}\right) \right\} \\
 &= E \left\{ (1 - p)^2 - 2(1 - p) \exp(-\eta V) + \exp(-2\eta V) \right\}, \tag{24}
 \end{aligned}$$

where

$$\eta = d / \widehat{\theta}. \tag{25}$$

Stage 2 Averaging of the decision criterion over the probability distribution (23) of the pivot V :

$$\begin{aligned}
 R(\theta, d) &= E \left\{ (1 - p)^2 - 2(1 - p) \exp(-\eta V) + \exp(-2\eta V) \right\} \\
 &= \int_0^{\infty} \left[(1 - p)^2 - 2(1 - p) \exp(-\eta v) + \exp(-2\eta v) \right] \varphi(v) dv \\
 &= \int_0^{\infty} \left[(1 - p)^2 - 2(1 - p) \exp(-\eta v) + \exp(-2\eta v) \right] \frac{n^n}{\Gamma(n)} v^{n-1} \exp(-nv) dv \\
 &= (1 - p)^2 - 2(1 - p) \frac{n^n}{(n + \eta)^n} + \frac{n^n}{(n + 2\eta)^n} \\
 &= (1 - p)^2 - 2(1 - p) \frac{1}{(1 + \eta/n)^n} + \frac{1}{(1 + 2\eta/n)^n}. \tag{26}
 \end{aligned}$$

Stage 3 Process of finding the optimal statistical decision rule:

If $p = 0.8, n = 2$ and $\widehat{\theta} = 10$, it can be shown that

$$\begin{aligned} \eta^* &= \arg \min_d R(\theta, d) \\ &= \arg \min_{\eta} \left((1-p)^2 - 2(1-p) \frac{1}{(1+\eta/n)^n} + \frac{1}{(1+2\eta/n)^n} \right) = 4.89598, \end{aligned} \quad (27)$$

The optimal estimator (statistical decision rule) d for estimating the quantile q is given by

$$d^* = \eta^* \hat{\theta} = 4.89598 \times 10 = 48.9598, \quad (28)$$

and the risk function is equal to

$$R(\theta, d^*) = E_{\theta} \{L(\theta, d^*)\} = 0.035121. \quad (29)$$

For comparison, the maximum likelihood estimator d_{ML} for estimating the quantile q is given by

$$d_{\text{ML}} = \hat{\vartheta} \hat{\theta} = -\ln(1-p) \hat{\theta} = 1.609438 \times 10 = 16.09438, \quad (30)$$

the risk function is equal to

$$R(\theta, d_{\text{ML}}) = E_{\theta} \{L(\theta, d_{\text{ML}})\} = 0.064049, \quad (31)$$

the relative efficiency of d_{ML} relative to d^* is given by

$$\text{rel. eff.}_R \{d_{\text{ML}}, d^* | \theta\} = \frac{R(\theta, d^*)}{R(\theta, d_{\text{ML}})} = \frac{0.035121}{0.064049} = 0.548346. \quad (32)$$

Thus, in this case, the use of d^* leads to a reduction in the risk of about 45.2% as compared with d_{ML} .

6 Characterization of Uniformly Non-dominated Decision Rules

A decision rule d is said to be uniformly non-dominated if there is no decision rule uniformly better than d . The conditions that a decision rule must satisfy in order that it might be uniformly non-dominated are given by the following theorem.

Theorem 1 (Uniformly non-dominated decision rule) Let $(\xi_{\tau}(\theta); \tau = 1, 2, \dots)$ be a sequence of the prior distributions on the parameter space Θ . Suppose that

$(d_\tau; \tau = 1, 2, \dots)$ and $(R^\bullet(\xi_\tau(\theta), d_\tau); \tau = 1, 2, \dots)$ are the sequences of Bayes decision rules and posterior risks, respectively. If there exists a statistical decision rule d such that its risk function $R(\theta, d), \theta \in \Theta$, satisfies the relationship

$$\lim_{\tau \rightarrow \infty} [R^\bullet(\xi_\tau(\theta), d) - R^\bullet(\xi_\tau(\theta), d_\tau)] = 0, \tag{33}$$

where a posterior risk (obtained through the posterior PDF $\xi_\tau^\bullet(\theta)$ of θ) is given by

$$R^\bullet(\xi_\tau(\theta), d) = \int_{\Theta} R(\theta, d) \xi_\tau^*(\theta) d\theta, \tag{34}$$

then d is a uniformly non-dominated decision rule.

Proof

Suppose d is uniformly dominated. Then there exists a decision rule d^* such that $R(\theta, d^*) < R(\theta, d)$ for all $\theta \in \Theta$. Let

$$\varepsilon = \inf_{\theta \in \Theta} [R(\theta, d) - R(\theta, d^*)] > 0. \tag{35}$$

Then

$$R^\bullet(\xi_\tau(\theta), d) - R^\bullet(\xi_\tau(\theta), d^*) \geq \varepsilon. \tag{36}$$

Simultaneously,

$$R^\bullet(\xi_\tau(\theta), d^*) - R^\bullet(\xi_\tau(\theta), d_\tau) \geq 0, \tag{37}$$

$\tau = 1, 2, \dots$, and

$$\lim_{\tau \rightarrow \infty} [R^\bullet(\xi_\tau(\theta), d^*) - R^\bullet(\xi_\tau(\theta), d_\tau)] \geq 0. \tag{38}$$

On the other hand,

$$R^\bullet(\xi_\tau(\theta), d^*) - R^\bullet(\xi_\tau(\theta), d_\tau) = [R^\bullet(\xi_\tau(\theta), d) - R^\bullet(\xi_\tau(\theta), d_\tau)]$$

$$- [R^\bullet(\xi_\tau(\theta), d) - R^\bullet(\xi_\tau(\theta), d^*)] \leq [R^\bullet(\xi_\tau(\theta), d) - R^\bullet(\xi_\tau(\theta), d_\tau)] - \varepsilon \tag{39}$$

and

$$\lim_{\tau \rightarrow \infty} [R^\bullet(\xi_\tau(\theta), d^*) - R^\bullet(\xi_\tau(\theta), d_\tau)] < 0. \tag{40}$$

This contradiction proves that d is a uniformly non-dominated decision rule.

Statistical Inference Using the result of Theorem 1, it can be shown that the optimal estimator (decision rule) d for estimating the quantile q , obtained through the proposed, in this chapter, the novel technique of invariant statistical embedding and averaging in terms of pivotal quantities (ISE&APQ), is uniformly non-dominated.

7 Example 2 (Constructing Shortest-Length Confidence Intervals)

If X_1, X_2, \dots, X_n is a random sample from a Pareto distribution with probability density function

$$f_\lambda(x) = \frac{\lambda}{x^2}, \quad x \geq \lambda, \quad \lambda > 0, \quad (41)$$

and cumulative distribution function

$$F_\lambda(x) = 1 - \frac{\lambda}{x}, \quad x \geq \lambda, \quad \lambda > 0, \quad (42)$$

where λ is a parameter, then what is the $100(1 - \alpha)\%$ shortest-length confidence interval for λ ?

Answer In this case,

$$X_{(1)} = \min(X_1, X_2, \dots, X_n) \quad (43)$$

is a complete sufficient statistic for λ with the probability density function (PDF)

$$q_\lambda(x_{(1)}) = \frac{n\lambda^n}{x_{(1)}^{n+1}}, \quad x_{(1)} \geq \lambda, \quad (44)$$

and the cumulative distribution function (CDF)

$$Q_\lambda(x_{(1)}) = 1 - \left(\frac{\lambda}{x_{(1)}}\right)^n, \quad x_{(1)} \geq \lambda. \quad (45)$$

The random variable

$$T = X_{(1)}/\lambda \quad (46)$$

has PDF

$$q(t) = n \left(\frac{1}{t}\right)^{n+1}, \quad t \geq 1, \tag{47}$$

and CDF

$$Q(t) = 1 - \left(\frac{1}{t}\right)^n, \quad t \geq 1. \tag{48}$$

Using T as pivot, we see that the confidence interval is $(X_{(1)}/b, X_{(1)}/a)$ with length

$$L = X_{(1)} (1/a - 1/b). \tag{49}$$

We minimize L subject to

$$\int_a^b q(t)dt = \int_a^b n \left(\frac{1}{t}\right)^{n+1} dt = Q(b) - Q(a) = \left(\frac{1}{a}\right)^n - \left(\frac{1}{b}\right)^n = 1 - \alpha. \tag{50}$$

Now

$$(1 - \alpha)^{1/n} < 1/a \leq 1 \quad \left(\text{or } 1 \leq a < 1/(1 - \alpha)^{1/n}\right) \tag{51}$$

and

$$\frac{dL}{da} = X_{(1)} \left(-\frac{1}{a^2} + \frac{1}{b^2} \frac{db}{da}\right) = X_{(1)} \left(\frac{b^{n-1} - a^{n-1}}{a^{n+1}}\right) > 0, \tag{52}$$

where

$$\frac{db}{da} = \frac{Q'(a)}{Q'(b)} = \frac{(1/a)^{n+1}}{(1/b)^{n+1}}, \tag{53}$$

so that the minimum occurs at $a = 1, b = 1/\alpha^{1/n}$. The shortest-length confidence interval for λ based on $X_{(1)}$ is $(X_{(1)}\alpha^{1/n}, X_{(1)})$. Note that

$$E \{L\} = E \{X_{(1)}\} \left(\frac{1}{a} - \frac{1}{b}\right) = \frac{n\lambda}{n-1} \left(\frac{1}{a} - \frac{1}{b}\right), \tag{54}$$

which is minimized subject to

$$(1/a)^n - (1/b)^n = 1 - \alpha, \tag{55}$$

where $a = 1$ and $b = 1/\alpha^{1/n}$. The expected length of the interval that minimizes $E\{L\}$ is $[1 - \alpha^{1/n}]n\lambda/(n - 1)$, which is also the expected length of the shortest confidence interval based on $X_{(1)}$. Note that the length of the interval $(X_{(1)}\alpha^{1/n}, X_{(1)})$ goes to 0 as $n \rightarrow \infty$.

The optimal numerical solution can be found using computer software ‘‘Solver’’ as follows:

Minimize

$$z = E\{L\} / E\{X_{(1)}\} = (1/a - 1/b) \tag{56}$$

subject to

$$(1/a)^n - (1/b)^n = 1 - \alpha, \tag{57}$$

$$1 \leq a < 1/(1 - \alpha)^{1/n}. \tag{58}$$

If, say, $n = 3$, $\alpha = 0.1$, it follows from the above that $a = 1$, $b = 2.154435 = 1/\alpha^{1/n}$, $z = 0.535841$.

8 Example 3 (Constructing Within-Sample Prediction Limits)

Let $Y_1 \leq \dots \leq Y_k$ be the first k ordered observations (order statistics) in a sample of size m from the exponential distribution with the probability density function

$$f_\theta(y) = \theta^{-1} \exp(-y/\theta), \quad \theta > 0, y > 0, \tag{59}$$

and the cumulative distribution function

$$F_\theta(y) = 1 - \exp(-y/\theta), \quad \bar{F}_\theta(y) = 1 - F_\theta(y), \quad \theta > 0, y > 0, \tag{60}$$

where θ is the scale parameter ($\theta > 0$). It is assumed that the parameter θ is unknown. In type II censoring, which is of primary interest here, the number of survivors are fixed and Y_k is a random variable. The MLE’s $\hat{\theta}$ of the parameter θ is given by

$$\hat{\theta} = \frac{S_k}{k} = \frac{\sum_{i=1}^k Y_i + (m - k) Y_k}{k}, \tag{61}$$

It is known that S_k is the complete sufficient statistic for θ . Then

$$V_k = S_k/\theta \tag{62}$$

is the pivotal quantity, the probability density function of which is given by

$$f(v_k) = \frac{1}{\Gamma(k)} v_k^{k-1} \exp(-v_k), \quad v_k \geq 0. \tag{63}$$

Theorem 2 (Exact lower statistical within-sample prediction limit with expected $(1 - \alpha)$ -confidence) The exact lower statistical within-sample prediction limit with expected $(1 - \alpha)$ -confidence, $L_l^\bullet \equiv L_l^\bullet(Y_k, S_k)$, on future outcomes of the l th order statistic Y_l from a set of m future ordered observations $Y_{k+1} \leq \dots \leq Y_m$ also from the distribution (59), which satisfies

$$E_\theta \{ \Pr(Y_l > L_l^\bullet | Y_k = y_k) \} = E_\theta \left\{ \int_{L_k^\bullet}^\infty g_\theta(y_l | y_k) dy_l \right\} = E_\theta \{ \bar{G}_\theta(L_l^\bullet | y_k) \} = 1 - \alpha, \tag{64}$$

where

$$g_\theta(y_l | y_k) = \frac{(m - k)!}{(l - k - 1)! (m - l)!} \times \left[1 - \frac{\bar{F}_\theta(y_l)}{\bar{F}_\theta(y_k)} \right]^{l-k-1} \left[\frac{\bar{F}_\theta(y_l)}{\bar{F}_\theta(y_k)} \right]^{m-l} \frac{f_\theta(y_l)}{\bar{F}_\theta(y_k)}, \quad y_l \geq y_k, \tag{65}$$

is the conditional probability density function of the l th order statistic Y_l , given that $Y_k = y_k$,

$$G_\theta(y_l | y_k) = \Pr(Y_l \leq y_l | Y_k = y_k) = \sum_{i=l-k}^{m-k} \binom{m-k}{i} \left[1 - \frac{\bar{F}_\theta(y_l)}{\bar{F}_\theta(y_k)} \right]^i \left[\frac{\bar{F}_\theta(y_l)}{\bar{F}_\theta(y_k)} \right]^{m-k-i} = \int_0^{1 - \frac{\bar{F}_\theta(y_l)}{\bar{F}_\theta(y_k)}} \frac{1}{B(l-k, m-l+1)} \tau^{l-k-1} (1-\tau)^{m-l} d\tau, \tag{66}$$

is given by

$$L_l^\bullet = Y_k + wS_k. \tag{67}$$

The value w of ancillary statistic $W = (Y_l - Y_k)/S_k$ is determined by

$$w = \arg \left(\frac{1}{B(l - k, m - l + 1)} \sum_{j=0}^{l-k-1} \binom{l - k - 1}{j} \right. \\ \left. \times \frac{(-1)^j}{m - l + 1 + j} [1 + (m - l + 1 + j) w]^{-k} = 1 - \alpha \right). \tag{68}$$

Proof The proof obtained using the proposed ISE & APQ technique is omitted here and will appear elsewhere.

Corollary 2.1 The exact upper statistical within-sample prediction limit with expected $(1 - \alpha)$ -confidence, $U_l^\bullet \equiv U_l^\bullet(Y_k, S_k)$, , on future outcomes of the l th order statistic Y_l from a set of m future ordered observations $Y_{k+1} \leq \dots \leq Y_m$ also from the distribution (59), which satisfies

$$E_\theta \{ \Pr(Y_l \leq U_l^\bullet | Y_k = y_k) \} = E_\theta \left\{ \int_0^{U_l^\bullet} g_\theta(y_l | y_k) dy_l \right\} = E_\theta \{ G_\theta(U_l^\bullet | y_k) \} = 1 - \alpha, \tag{69}$$

is given by

$$U_l^\bullet = Y_k + w S_k, \tag{70}$$

where the value w of ancillary statistic $W = (Y_l - Y_k)/S_k$ is determined by

$$w = \arg \left(\frac{1}{B(l - k, m - l + 1)} \sum_{j=0}^{l-k-1} \binom{l - k - 1}{j} \right. \\ \left. \times \frac{(-1)^j}{m - l + 1 + j} [1 + (m - l + 1 + j) w]^{-k} = \alpha \right). \tag{71}$$

Theorem 3 (Exact lower statistical within-sample prediction limit with expected $(1 - \alpha)$ -confidence) Let us assume that the first $k-1$ ordered observations (order statistics) $Y_1 \leq \dots \leq Y_{k-1}$ in a sample of size m from the exponential distribution with the probability density function (59) are unknown. Then the exact lower statistical within-sample prediction limit with expected $(1 - \alpha)$ -confidence, $L_l^\bullet \equiv$

$L_l^\bullet(Y_k)$, on future outcomes of the l th order statistic Y_l from a set of m future ordered observations $Y_{k+1} \leq \dots \leq Y_m$ also from the distribution (59), which satisfies (64), is given by

$$L_l^\bullet = (1 + w_1) Y_k, \tag{72}$$

where the value w_1 of ancillary statistic $W_1 = (Y_l - Y_k)/Y_k$ is determined by

$$w_1 = \arg \left(\frac{\frac{m!}{(m-l)!(l-k-1)!} \sum_{j=0}^{l-k-1} \binom{l-k-1}{j} \frac{(-1)^j}{m-l+1+j}}{\times \left[\prod_{i=0}^{k-1} (w_1(m-l+1+j) + m-k+1+i) \right]^{-1}} = 1 - \alpha \right). \tag{73}$$

Proof The proof obtained via the proposed ISE & APQ technique is omitted here and will appear elsewhere.

Corollary 3.1 The exact upper statistical within-sample prediction limit with expected $(1 - \alpha)$ -confidence, $U_l^\bullet \equiv U_l^\bullet(Y_k)$, on future outcomes of the l th order statistic Y_l from a set of m future ordered observations $Y_{k+1} \leq \dots \leq Y_m$ also from the distribution (59), which satisfies (69), is given by

$$U_l^\bullet = (1 + w_1) Y_k, \tag{74}$$

where the value w_1 of ancillary statistic $W_1 = (Y_l - Y_k)/Y_k$ is determined by

$$w_1 = \arg \left(\frac{\frac{m!}{(m-l)!(l-k-1)!} \sum_{j=0}^{l-k-1} \binom{l-k-1}{j} \frac{(-1)^j}{m-l+1+j}}{\times \left[\prod_{i=0}^{k-1} (w_1(m-l+1+j) + m-k+1+i) \right]^{-1}} = \alpha \right). \tag{75}$$

Theorem 4 (Exact lower statistical within-sample γ -content prediction limit with expected $(1 - \alpha)$ -confidence) The exact lower statistical within-sample prediction limit with expected $(1 - \alpha)$ -confidence, $L_l^{\bullet\bullet} \equiv L_l^{\bullet\bullet}(Y_k, S_k)$, on future outcomes of the l th order statistic Y_l from a set of m future ordered observations $Y_{k+1} \leq \dots \leq Y_m$ also from the distribution (59), which satisfies

$$E_{\theta} \left\{ \Pr \left(\int_{L_l^{**}}^{\infty} g_{\theta}(y_l | y_k) dy_l \geq \gamma \right) \right\} = E_{\theta} \{ \Pr (\bar{G}_{\theta}(L_l^{**} | y_k) \geq \gamma) \} = 1 - \alpha, \tag{76}$$

is given by

$$L_l^{**} = Y_k + \eta_l S_k. \tag{77}$$

The value of the ancillary factor η_l is determined by

$$\eta_l = \arg \left(\int_0^{\infty} \left(\int_0^{\ln(1-q_{1-\gamma})^{-1} v_k^{-1} - \eta_l} g(w) dw \right) f(v_k) dv_k = 1 - \alpha \right), \tag{78}$$

where

$$g(w) = \frac{1}{B(l-k, m-l+1)} \sum_{j=0}^{l-k-1} \binom{l-k-1}{j} (-1)^j \times k[1 + (m-l+1+j)w]^{-k-1}, \quad w \in (0, \infty). \tag{79}$$

Proof The proof obtained using the proposed ISE&APQ technique is omitted here and will appear elsewhere.

Theorem 5 (Exact lower statistical within-sample γ -content prediction limit with expected $(1 - \alpha)$ -confidence) Let us assume that the first $k - 1$ ordered observations (order statistics) $Y_1 \leq \dots \leq Y_{k-1}$ in a sample of size m from the exponential distribution with the probability density function (59) are unknown. Then the exact lower statistical within-sample prediction limit with expected $(1 - \alpha)$ -confidence, $L_l^{**} \equiv L_l^{**}(Y_k)$, on future outcomes of the l th order statistic Y_l from a set of m future ordered observations $Y_{k+1} \leq \dots \leq Y_m$ also from the distribution (59), which satisfies (76), is given by

$$L_l^{**} = (1 + \eta_l) Y_k. \tag{80}$$

The value of the ancillary factor η_l is determined by

$$\eta_l = \arg \left(\int_0^\infty \left(\int_0^{\ln(1-q_{1-\gamma})^{-1}v_1^{-1}-\eta_l} g_1(w_1) dw_1 \right) f_1(v_1) dv_1 = 1 - \alpha \right), \tag{81}$$

where

$$g_1(w_1) = \frac{m!}{(m-l)!(l-k-1)!(k-1)!} \sum_{j=0}^{l-k-1} \binom{l-k-1}{j} (-1)^j$$

$$\times \sum_{i=0}^{k-1} \binom{k-1}{i} (-1)^i \frac{1}{[w_1(m-l+1+j) + m-k+1+i]^2}, \quad w_1 \in (0, \infty), \tag{82}$$

$$V_1 = Y_k/\theta, \tag{83}$$

$$f_1(v_1) = \frac{1}{B(k, m-k+1)} \sum_{i=0}^{k-1} \binom{k-1}{i} (-1)^i \exp(-v_1(m-k+1+i)), \quad v_1 \in (0, \infty). \tag{84}$$

Proof The proof obtained via the proposed ISE & APQ technique is omitted here and will appear elsewhere.

Remark The exact upper statistical γ -content prediction limit with expected $(1 - \alpha)$ -confidence, $U_l^{\bullet\bullet}$, can be obtained from the exact lower statistical γ -content prediction limit with expected $(1 - \alpha)$ -confidence, $L_l^{\bullet\bullet}$, by replacing γ by $1 - \gamma$, and $1 - \alpha$ by α .

9 Conclusion

In statistics, a pivotal quantity or pivot is a function of observations and unobservable parameters such that the function’s probability distribution does not depend on the unknown parameters (including nuisance parameters). A pivotal quantity need not be a statistic – the function and its value can depend on the parameters of the model, but its distribution must not. If it is a statistic, then it is known as an ancillary statistic.

The new intelligent analytical technique of ISE&APQ proposed in this chapter represents the conceptually simple, efficient, and useful method for constructing exact, optimal, or improved statistical decision rules under parametric uncertainty

of underlying models. For example, we construct the following one-sided statistical (within-sample or new-sample) prediction limits: (i) one-sided statistical prediction limit that covers at least $100\gamma\%$ of the observations with expected $100(1 - \alpha)\%$ confidence, (ii) one-sided statistical prediction limit determined so that the expected proportion of the observations covered by this limit is $(1 - \alpha)$. Such prediction limits are required, when planning life tests, engineers may need to predict the number of failures that will occur by the end of the test or to predict the amount of time that it will take for a specified number of units to fail, and so on.

The methodology described here can be extended in several different directions to solve various problems arising in practice.

References

1. N.A. Nechval, E.K. Vasermanis, *Improved Decisions in Statistics* (Izglitibas soli, Riga, 2004)
2. N.A. Nechval, G. Berzins, M. Purgailis, K.N. Nechval, Improved estimation of state of stochastic systems via invariant embedding technique. *WSEAS Trans. Math.* **7**, 141–159 (2008)
3. N.A. Nechval, K.N. Nechval, V. Danovich, T. Liepins, Optimization of new-sample and within-sample prediction intervals for order statistics, in *Proceedings of the 2011 World Congress in Computer Science, Computer Engineering, and Applied Computing*, WORLDCOMP'11, Las Vegas Nevada, USA, CSREA Press, July 18–21, 2011, pp. 91–97
4. N.A. Nechval, K.N. Nechval, G. Berzins, A new technique for intelligent constructing exact γ -content tolerance limits with expected $(1 - \alpha)$ -confidence on future outcomes in the Weibull case using complete or type II censored data. *Autom. Control Comput. Sci. (AC&CS)* **52**, 476–488 (2018)
5. N.A. Nechval, G. Berzins, K.N. Nechval, Intelligent technique of constructing exact statistical tolerance limits to predict future outcomes under parametric uncertainty for prognostics and health management of complex systems. *Int. J. Adv. Comput. Sci. Appl. (IJCSIA)* **9**, 30–47 (2019)
6. R.H. Berk, A special group structure and equivariant estimation. *Ann. Math. Stat.* **38**, 1436–1446 (1967)
7. T.S. Ferguson, *Mathematical Statistics* (Academic Press, New York, 1967)

A Note on the Sensitivity of Generic Approximate Sparse Pseudoinverse Matrix for Solving Linear Least Squares Problems



A. D. Lipitakis, G. A. Gravvanis, C. K. Filelis-Papadopoulos,
and D. Anagnostopoulos

1 Introduction

Let us consider a sparse linear least squares problem of the following form [6]:

$$\min_x \|b - Ax\|_2 \quad (1)$$

where $A \in \mathbb{R}^{m \times n}$, with $m > n$, is a large sparse coefficient matrix, $b \in \mathbb{R}^{m \times 1}$ is the right-hand side vector, and $x \in \mathbb{R}^{n \times 1}$ is the solution vector. The solution of the problem (1) is equivalent to solving the normal equations:

$$A^T A X = A^T b \quad (2)$$

However, the condition number of the coefficient matrix of the square sparse linear system (2) is $\kappa(A^T A) = \kappa(A)^2$.

Direct and iterative methods can be used for solving least squares problem (1). Several direct methods have been introduced such as Householder QR [21] and Multifrontal Sparse QR [14]. Preconditioned iterative methods, using effective

A. D. Lipitakis (✉) · D. Anagnostopoulos
Department of Informatics and Telematics, Harokopio University of Athens, Athens, Greece
e-mail: adlipita@hua.gr; dimosthe@hua.gr

G. A. Gravvanis
Department of Electrical and Computer Engineering, School of Engineering, Democritus
University of Thrace, University Campus, Xanthi, Greece
e-mail: ggravvan@ee.duth.gr

C. K. Filelis-Papadopoulos
Western Gateway Building, University College Cork, Cork, Ireland
e-mail: christos.papadopoulos-filelis@cs.ucc.ie

preconditioners, have been used for solving least squares problems arising in various scientific fields [3, 4, 6, 8, 22, 27, 29, 30].

Common approaches in the literature include SSOR-based preconditioning [6] and preconditioners using projection and block projection of rows or columns [7, 8]. Incomplete QR factorization such as rTIGO, cTIGO [27] and related variants [3, 22, 26, 29, 30] has also been used as preconditioners in conjunction with iterative methods. In addition, the robust incomplete factorization [4, 23] was based on the A-orthogonalization technique for the computation of the root-free Cholesky factorization, avoiding the explicit formation of $A^T A$. The multilevel incomplete QR (MIQR) method has been used as a preconditioner with the CGLS method [24]. Moreover, a balanced incomplete factorization (BIF) technique using approximate inverse (AINV) decomposition techniques has been considered for preconditioning linear systems [9], while non-singular basis matrices for preconditioning large linear systems have been also proposed in [1].

A major disadvantage of factorization-based methods is that they have limited parallelism. Explicit approximate inverses, which possess inherent parallelism, based on approximate or incomplete factorizations have been proposed [12, 13, 16, 26]. A family of classes of explicit approximate sparse inverses in conjunction with approximate inverse sparsity patterns [10] has been proposed for various model problems [12, 26].

Various techniques for sensitivity analysis of linear least squares problems have been widely studied [2, 5]. Furthermore, the sensitivity of explicit approximate inverses and the convergence rate for solving sparse linear systems, arising from the numerical solution of partial differential equations, have been examined, [13, 15, 18, 19].

This paper is organized as follows: In Sect. 2, the explicit preconditioned approach, based on generic approximate sparse pseudoinverses, for solving least squares problems is given. In Sect. 3, the generic approximate sparse pseudoinverses based on modified row-threshold incomplete QR factorization techniques are presented. In Sect. 4, a set of modified Moore-Penrose conditions, based on generic approximate sparse pseudoinverse (GASP) technique and theoretical estimates for the sensitivity of GASP are given. Finally, in Sect. 5, numerical results of the proposed method for solving various problems are presented. The theoretical estimates of the approximate pseudoinverse were found to be in qualitative agreement with the numerical results.

2 Explicit Preconditioned Conjugate Gradient Least Squares Method

The explicit preconditioning methods have been effectively used for solving large linear least square problems. The Explicit Preconditioned Conjugate Gradient Least Squares (EPCGLS) method [6, 25, 26], in conjunction with generic approximate

sparse pseudoinverse (GASP) scheme, can then be described by the following algorithm:

Algorithm 1 EPCGLS algorithm

```

1: Given an initial approximation  $x_0$ 
2: Compute  $r_0 = b - Ax_0$ 
3: Compute  $z_0 = \tilde{M}_{droptol}^{lfill} r_0 = (G_{droptol}^{lfill})(G_{droptol}^{lfill})^T A^T r_0$ 
4: Set  $p_0 = z_0$ 
5: For  $i=0, \dots, N_{max}$ 
6:   Compute  $w_i = Ap_i$ 
7:   Compute  $a_i = \frac{(z_i, A^T r_i)}{\|w_i\|_2^2}$ 
8:   Compute  $x_{i+1} = x_i + a_i p_i$ 
9:   Compute  $r_{i+1} = r_i - a_i w_i$ 
10:  If  $\|A^T r_{i+1}\| < \varepsilon \|A^T r_0\|$  terminate
11:  Compute
12:     $z_{i+1} = \tilde{M}_{droptol}^{lfill} r_{i+1} = (G_{droptol}^{lfill})(G_{droptol}^{lfill})^T A^T r_{i+1}$ 
13:  Compute  $\beta_i = \frac{(z_{i+1}, A^T r_{i+1})}{(z_i, A^T r_i)}$ 
14:  Compute  $p_{i+1} = z_{i+1} + \beta_i p_i$ 
15: End for
16:

```

where N_{max} denotes the maximum number of iterations and ε the prescribed tolerance. The computational complexity of the EPCGLS method in conjunction with the GASP matrix is $\approx O((6nnz(A) + 8n + 6m + 4nnz(G_{droptol}^{lfill}))\nu)$, where ν denotes the number of required iterations for convergence to the prescribed tolerance. The explicit approximate inverses have been efficiently parallelized, and the implementation details concerning the parallelization techniques have been extensively presented in [13, 16, 17].

3 Generic Approximate Sparse Pseudoinverse Scheme

Let us consider the following sparse linear system:

$$Ax = b \quad (3)$$

where A is the $(m \times n)$ coefficient matrix with full column rank, x is the solution vector, and b is the right-hand side vector. The coefficient matrix A can be factorized as follows:

$$A = \tilde{Q}\tilde{R} + E \quad (4)$$

where the factor \tilde{Q} is a $(m \times n)$ sparse orthogonal matrix, the factor \tilde{R} is a $(n \times n)$ sparse upper triangular matrix, and E is the $(m \times n)$ error matrix.

The use of Givens rotations retains the \tilde{Q} matrix orthogonal [3, 27], while simultaneously its explicit formation is avoided. Note that \tilde{Q} is computed as a sequence of Givens rotations. The approximate decomposition factors \tilde{Q} and \tilde{R} are computed using the mrTIGO method [26]. Note that the factor \tilde{Q} is not stored in order to reduce the memory requirements.

To reduce the memory requirements, several techniques are used including the class of Q-less QR decomposing methods [14, 27], which do not form the matrix Q explicitly. For very large linear systems, the number of fill-in terms stored in the R factor may be of substantial order. In order to avoid computation of fill-in terms and enhance parallelization during the solution of these systems, preconditioned iterative methods can be used, which rely on the use of effective preconditioning schemes [6].

The GASP matrix is in factored form and is based on modified row-wise incomplete Q-less QR factorization (mrTIGO) scheme [26]. This scheme is based on Givens rotations and a modified filtering procedure for admitting fill-in terms computed during the factorization process. The orthogonal matrix Q computed as the product of Givens rotations is discarded in order to reduce memory requirements. The approximate pseudoinverse can be computed as follows:

$$\begin{aligned} M &= ((\tilde{Q}\tilde{R} + E)^T(\tilde{Q}\tilde{R} + E))^{-1}A^T \\ &= (\tilde{R}^T\tilde{Q}^T\tilde{Q}\tilde{R} + E^T\tilde{Q}\tilde{R} + \tilde{R}^T\tilde{Q}^TE + E^TE)^{-1}A^T = \tilde{M} + \hat{M} \end{aligned} \quad (5)$$

with

$$\tilde{M} = (\tilde{R}^T\tilde{Q}^T\tilde{Q}\tilde{R})^{-1}A^T = (\tilde{R}^T\tilde{R})^{-1}A^T = \tilde{G}\tilde{G}^TA^T \quad (6)$$

where \tilde{M} denotes the approximate pseudoinverse matrix, \hat{M} the error matrix, and $G = \tilde{R}^{-1}$ is the inverse of the sparse upper triangular factor obtained through the modified incomplete QR factorization [26]. In order to reduce memory requirements, the sparse approximate inverse matrix $\tilde{G}_{droptol}^{lfill} = \tilde{R}^{-1}$ is computed based on a sparsity pattern formed by the lfill power of a sparsified version of the matrix R . The sparsification of matrix R is performed based on a predefined drop tolerance, namely, droptol [10]. Thus, the sparse approximate pseudoinverse matrix is formed as follows:

$$\tilde{M}_{droptol}^{lfill} = (\tilde{G}_{droptol}^{lfill})(\tilde{G}_{droptol}^{lfill})^TA^T \quad (7)$$

In the following section, conditions for estimating the sensitivity of the aforementioned explicit preconditioning scheme are presented.

4 On Theoretical Estimates for the Sensitivity of the Generic Sparse Approximate Pseudoinverse Matrix

In this section, we present modified Moore-Penrose conditions and theoretical estimates concerning the classes of the generic approximate sparse pseudoinverse matrix.

The Moore-Penrose conditions [28] refer to the quality of the pseudoinverse matrix, which are

$$(i)AMA = A, (ii)MAM = M, (iii)(AM)^T = AM, (iv)(MA)^T = MA \quad (8)$$

Additionally, the modified Moore-Penrose conditions [28], based on the various classes of the GASP matrix, are presented. These conditions are then written as follows:

$$\begin{aligned} (i) \tilde{M}_{droptol}^{lfill}A &= A & (iii) (A\tilde{M}_{droptol}^{lfill})^T &= A\tilde{M}_{droptol}^{lfill} \\ (ii) \tilde{M}_{droptol}^{lfill}A\tilde{M}_{droptol}^{lfill} &= \tilde{M}_{droptol}^{lfill} & (iv) (\tilde{M}_{droptol}^{lfill}A)^T &= \tilde{M}_{droptol}^{lfill}A \end{aligned}$$

In the following, we present theoretical estimates on the sensitivity of the approximate inverse matrix.

It is known that if $A \in \mathbb{R}^{m \times n}$ with $m \gg n$ ($A \neq 0$), then $\kappa(A) = \|A\|_2 \|\tilde{M}\|_2 = \sigma_1 / \sigma_r$, where $\sigma_1 > \sigma_2 > \dots > \sigma_r > 0$ are nonzero singular values.

In order to derive an upper bound for

$$\Omega_{droptol}^{lfill} = \tilde{M}_{droptol}^{lfill}A \quad (9)$$

we have

$$\Omega_{droptol}^{lfill} = (M - \tilde{M}_{error})M^{-1} = MA - \tilde{M}_{error}M^{-1} = I - \tilde{M}_{error}M^{-1}. \quad (10)$$

By taking norm, we obtain

$$\|\Omega_{droptol}^{lfill}\| \leq 1 + \|\tilde{M}_{error}\| \|M^{-1}\| \leq 1 + \kappa(A) \frac{\|\tilde{M}_{error}\|}{\|M\|} \quad (11)$$

By definition we have

$$\|\tilde{M}_{droptol}^{lfill}\| \leq \|M\|. \quad (12)$$

Hence,

$$\| \Omega_{droptol}^{lfill} \| \leq 1 + \kappa(A) \frac{\| \tilde{M}_{error} \|}{\| \tilde{M}_{droptol}^{lfill} \|} \quad (13)$$

In order to derive a lower bound for

$$\Omega_{droptol}^{lfill} = \tilde{M}_{droptol}^{lfill} A \quad (14)$$

we obtain the following

$$\begin{aligned} \Omega_{droptol}^{lfill} &= \tilde{M}_{droptol}^{lfill} A = \tilde{M}_{droptol}^{lfill} [\tilde{M}_{droptol}^{lfill} + \tilde{M}_{error}]^{-1} \\ &= [(\tilde{M}_{droptol}^{lfill} + \tilde{M}_{error})(\tilde{M}_{droptol}^{lfill})^{-1}]^{-1} \end{aligned} \quad (15)$$

Then, by taking norms, we have

$$\begin{aligned} \| \Omega_{droptol}^{lfill} \| &= \| [(\tilde{M}_{droptol}^{lfill} + \tilde{M}_{error})(\tilde{M}_{droptol}^{lfill})^{-1}]^{-1} \| \\ &\geq \frac{1}{\| (\tilde{M}_{droptol}^{lfill} + \tilde{M}_{error})(\tilde{M}_{droptol}^{lfill})^{-1} \|} \\ &\geq \frac{1}{\| \tilde{M}_{droptol}^{lfill} + \tilde{M}_{error} \| \| (\tilde{M}_{droptol}^{lfill})^{-1} \|} \\ &\geq \frac{1}{(\| \tilde{M}_{droptol}^{lfill} \| + \| \tilde{M}_{error} \|) \| \tilde{M}_{droptol}^{lfill} \|^{-1}} \\ &\geq \frac{1}{(\| M \| + \| \tilde{M}_{error} \|) \| M \|^{-1}} \geq \frac{1}{\kappa(A)(1 + \frac{\| \tilde{M}_{error} \|}{\| M \|})} \end{aligned} \quad (16)$$

It is evident that in the case of an exact inverse, $\| \tilde{M}_{error} M^{-1} \| = \| \tilde{M}_{error} A \| \rightarrow 0$, and thus $\| MA \| \rightarrow 1$.

5 Numerical Results

In this section, we examine the applicability of the proposed method and the quality of GASP technique. Numerical experiments have been performed on the ARIS supercomputer (GRNET). Each compute node consists of $2 \times$ Ivy Bridge Intel Xeon E5-2680v2 (10 cores each) and 64 GB RAM.

In Table 1, the characteristics of two model problems (*illc1850* refers to least squares in a surveying problem and *well1033* refers to unsymmetric least squares problems), were obtained from the University of Florida sparse matrix collection

[11], and the characteristics of a model problem (*small*) referring to animal breeding studies are given by Hegland [20].

It should be mentioned that the convergence behavior of the proposed preconditioning scheme, namely, Explicit Preconditioned Conjugate Gradient Least Squares (EPCGLS) method, based on the GASP scheme, is better than the convergence behavior of the Implicit Preconditioned Conjugate Gradient Least Squares (IPCGLS) method, in conjunction with incomplete Cholesky factorization with zero fill-in applied to the normal equations, and is significantly improved compared to the convergence behavior of the Conjugate Gradients Least Squares (CGLS) method applied to the normal equations for various model problems [26]. Additionally, the performance of the EPCGLS method, based on the GASP scheme for various model problems along with comparison to existing methods, is given in [26], while the performance of the parallel GASP scheme, given in [25], is substantially improved.

In Tables 2, 4, and 6, computed error estimates for the modified Moore-Penrose conditions in conjunction with the GASP matrix, for the model problems given in Table 1, with various values of the parameter *lfill* are shown. Computed relative error estimates for the modified Moore-Penrose conditions based on the GASP matrix, for the model problems given in Table 1, with various values of the parameter *lfill* are presented in Tables 3, 5, and 7. Additionally, in Table 8, computed error estimates of the $\| I - \tilde{M}_{droptol}^{lfill} A \|_2$ for the model problems with various values of the parameter *lfill* are given. Moreover, in Table 9, computed error estimates of the modified Moore-Penrose conditions applied to the normal equation, based on the exact inverse matrix, namely, $(M = (A^T A)^{-1} A^T)$, for various model problems are shown. Furthermore, in Table 10, computed relative error estimates for the modified Moore-Penrose conditions, based on the exact inverse matrix, namely, $(M = (A^T A)^{-1} A^T)$, for various model problems are presented. These estimates are used to evaluate the estimates of Tables 2, 3, 4, 5, 6, and 7, acting as experimental bounds to the computed norms, since for large *lfill* values the approximate inverse should asymptotically tend to the exact inverse. For denser approximate inverses, the improvement on the values of the norms toward the values corresponding to those of the exact inverse slows down, since the larger elements, which contribute more, are usually captured with smaller values of the *lfill* parameter.

In Figs. 1a, 2a, and 3a, the L_2 -norms of the error estimates against the number of iterations of the EPCGLS method, based on GASP scheme, with *lfill* = 2 and *lfill* = 4 for the model problems of Table 1 are presented. In Figs. 1b, 2b, and 3b, the L_2 -norms of the residuals against the number of iterations of the EPCGLS

Table 1 Model problems and their characteristics

Model problem	Rows	Columns	nnz(A)	nnz(A ^T A)	κ(A)
<i>illc1850</i>	1850	712	8636	9060	1.405E+03
<i>well1033</i>	1033	320	4732	3934	1.66E+02
<i>small</i>	3140	1988	8510	14532	3.9E+05

Table 2 Computed error estimates for *illc* 1850 model problem

Norms	τ	<i>droptol</i>	<i>lfill</i> = 1	<i>lfill</i> = 2	<i>lfill</i> = 3	<i>lfill</i> = 4	<i>lfill</i> = 8	<i>lfill</i> = 12
$\ \tilde{A}_{droptol}^{jfill} A - A \ _2$	0.001	0.01	6.35E+02	1.81E+01	1.81E+01	1.81E+01	6.79E-01	6.79E-01
$\ \tilde{M}_{droptol}^{jfill} A \tilde{M}_{droptol}^{jfill} - \tilde{M}_{droptol}^{jfill} \ _2$	0.001	0.01	2.44E+06	2.32E+03	4.55E+02	4.46E+02	4.45E+02	4.45E+02
$\ (A \tilde{M}_{droptol}^{jfill})^T - A \tilde{M}_{droptol}^{jfill} \ _2$	0.001	0.01	7.16E-13	1.13E-13	1.24E-13	1.14E-13	1.23E-13	1.11E-13
$\ (\tilde{M}_{droptol}^{jfill} A)^T - \tilde{M}_{droptol}^{jfill} A \ _2$	0.001	0.01	1.11E+04	3.26E+02	2.46E+02	2.39E+02	2.36E+02	2.36E+02

Table 3 Computed relative error for *illc* 1850 model problem

Norms	τ	<i>droptol</i>	<i>lfill</i> =1	<i>lfill</i> =2	<i>lfill</i> =3	<i>lfill</i> =4	<i>lfill</i> =8	<i>lfill</i> =12
$\ A\tilde{M}_{droptol}^{lfill}A - A\ _2 / \ A\ _2$	0.001	0.01	2.99E+02	8.52E+00	6.09E-01	3.21E-01	3.20E-01	3.20E-01
$\ \tilde{M}_{droptol}^{lfill}A\tilde{M}_{droptol}^{lfill} - \tilde{M}_{droptol}^{lfill}\ _2 / \ \tilde{M}_{droptol}^{lfill}\ _2$	0.001	0.01	3.73E+02	3.26E+00	6.47E-01	6.38E-01	6.36E-01	6.36E-01
$\ (A\tilde{M}_{droptol}^{lfill})^T - A\tilde{M}_{droptol}^{lfill}\ _2 / \ A\tilde{M}_{droptol}^{lfill}\ _2$	0.001	0.01	1.92E-15	7.63E-15	5.81E-14	5.91E-14	6.41E-14	5.77E-14
$\ (\tilde{M}_{droptol}^{lfill}A)^T - \tilde{M}_{droptol}^{lfill}A\ _2 / \ \tilde{M}_{droptol}^{lfill}A\ _2$	0.001	0.01	9.99E-01	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00

Table 4 Computed error estimates for *wel/1033* model problem

Norms	τ	<i>d</i> _{optol}	<i>l</i> _{fill} =1	<i>l</i> _{fill} =2	<i>l</i> _{fill} =3	<i>l</i> _{fill} =4	<i>l</i> _{fill} =8	<i>l</i> _{fill} =12
$\ \tilde{A}^{J_{fill}}_{droptol} A - A \ _2$	0.001	0.01	6.42E+02	1.31E-01	1.31E-01	1.32E-01	1.32E-01	1.32E-01
$\ \tilde{M}^{J_{fill}}_{droptol} A \tilde{M}^{J_{fill}}_{droptol} - \tilde{M}^{J_{fill}}_{droptol} \ _2$	0.001	0.01	6.50E+05	1.22E+01	1.22E+01	1.23E+01	1.23E+01	1.23E+01
$\ (A \tilde{M}^{J_{fill}}_{droptol})^T - A \tilde{M}^{J_{fill}}_{droptol} \ _2$	0.001	0.01	2.12E-13	1.46E-14	1.70E-14	1.68E-14	1.58E-14	1.58E-14
$\ (\tilde{M}^{J_{fill}}_{droptol} A)^T - \tilde{M}^{J_{fill}}_{droptol} A \ _2$	0.001	0.01	1.62E+03	1.11E+01	1.10E+01	1.11E+01	1.11E+01	1.11E+01

Table 5 Computed relative error for *well1033* model problem

Norms	τ	<i>d</i> optol	<i>l</i> fill = 1	<i>l</i> fill = 2	<i>l</i> fill = 3	<i>l</i> fill = 4	<i>l</i> fill = 8	<i>l</i> fill = 12
$\ A \tilde{M}_{droptol}^{jfill} A - A \ _2 / \ A \ _2$	0.001	0.01	3.56E+02	7.27E-02	7.26E-02	7.29E-02	7.30E-02	7.30E-02
$\ \tilde{M}_{droptol}^{jfill} A \tilde{M}_{droptol}^{jfill} - \tilde{M}_{droptol}^{jfill} \ _2 / \ \tilde{M}_{droptol}^{jfill} \ _2$	0.001	0.01	4.90E+02	1.30E-01	1.30E-01	1.31E-01	1.31E-01	1.31E-01
$\ (A \tilde{M}_{droptol}^{jfill})^T - A \tilde{M}_{droptol}^{jfill} \ _2 / \ A \tilde{M}_{droptol}^{jfill} \ _2$	0.001	0.01	4.19E-16	1.27E-14	1.48E-14	1.46E-14	1.37E-14	1.37E-14
$\ (\tilde{M}_{droptol}^{jfill} A)^T - \tilde{M}_{droptol}^{jfill} A \ _2 / \ \tilde{M}_{droptol}^{jfill} A \ _2$	0.001	0.01	9.57E-01	9.92E-01	9.92E-01	9.92E-01	9.92E-01	9.92E-01

Table 6 Computed error estimates for *small* model problem (animal breeding)

Norms	τ	<i>d</i> _{droptol}	<i>l</i> ^{fill} = 1	<i>l</i> ^{fill} = 2	<i>l</i> ^{fill} = 3	<i>l</i> ^{fill} = 4	<i>l</i> ^{fill} = 8	<i>l</i> ^{fill} = 12
$\ A \tilde{M}_{droptol}^{fill} A - A \ _2$	0.1	0.1	4.29E+04	5.33E+02	2.11E+00	1.62E+00	1.55E+00	1.55E+00
$\ \tilde{M}_{droptol}^{fill} A \tilde{M}_{droptol}^{fill} - \tilde{M}_{droptol}^{fill} \ _2$	0.1	0.1	7.73E+07	3.62E+05	4.51E+01	4.49E+01	4.49E+01	4.49E+01
$\ (A \tilde{M}_{droptol}^{fill})^T - A \tilde{M}_{droptol}^{fill} \ _2$	0.1	0.1	1.24E-12	2.82E-13	9.95E-15	1.01E-14	1.01E-14	1.02E-14
$\ (\tilde{M}_{droptol}^{fill} A)^T - \tilde{M}_{droptol}^{fill} A \ _2$	0.1	0.1	6.19E+04	5.08E+03	2.93E+02	2.93E+02	2.93E+02	2.93E+02

Table 7 Computed relative error estimates for *small* model problem (animal breeding)

Norms	τ	<i>droptol</i>	<i>lfill</i> = 1	<i>lfill</i> = 2	<i>lfill</i> = 3	<i>lfill</i> = 4	<i>lfill</i> = 8	<i>lfill</i> = 12
$\ A\tilde{M}_{droptol}^{lfill}A - A\ _2 / \ A\ _2$	0.1	0.1	2.53E+03	3.15E+01	1.24E-01	9.58E-02	9.16E-02	9.16E-02
$\ \tilde{M}_{droptol}^{lfill}A\tilde{M}_{droptol}^{lfill} - \tilde{M}_{droptol}^{lfill}\ _2 / \ \tilde{M}_{droptol}^{lfill}\ _2$	0.1	0.1	7.28E+03	1.95E+02	4.85E-01	4.83E-01	4.83E-01	4.83E-01
$\ (A\tilde{M}_{droptol}^{lfill})^T - A\tilde{M}_{droptol}^{lfill}\ _2 / \ A\tilde{M}_{droptol}^{lfill}\ _2$	0.1	0.1	1.70E-16	1.44E-15	6.69E-15	6.81E-15	6.83E-15	6.85E-15
$\ (\tilde{M}_{droptol}^{lfill}A)^T - \tilde{M}_{droptol}^{lfill}A\ _2 / \ \tilde{M}_{droptol}^{lfill}A\ _2$	0.1	0.1	9.93E-01	9.99E-01	1.00E+00	1.00E+00	1.00E+00	1.00E+00

Table 8 Computed error estimates of $\| I - \tilde{M}_{droptol}^{lfill} A \|_2$ for various model problems

model problem	τ	<i>droptol</i>	<i>lfill</i> = 1	<i>lfill</i> = 2	<i>lfill</i> = 3	<i>lfill</i> = 4	<i>lfill</i> = 8	<i>lfill</i> = 12
<i>illc1850</i>	0.001	0.01	1.11E+04	3.26E+02	2.46E+02	2.39E+02	2.36E+02	2.36E+02
<i>well1033</i>	0.001	0.01	1.69E+03	1.11E+01	1.10E+01	1.11E+01	1.11E+01	1.11E+01
<i>small</i>	0.1	0.1	6.23E+04	5.08E+03	2.93E+02	2.93E+02	2.93E+02	2.93E+02

Table 9 Computed error estimates using exact inverse for various problems

Norms using exact inverse ($M = (A^T A)^{-1} A^T$)	<i>illc1850</i>	<i>well1033</i>	<i>small</i>
$\ AMA - A \ _2$	1.06E-10	7.32E-13	1.02E+00
$\ MAM - M \ _2$	5.92E-08	1.74E-10	2.05E+06
$\ (AM)^T - AM \ _2$	3.34E-11	6.33E-13	7.57E-02
$\ (MA)^T - MA \ _2$	7.82E-09	3.54E-12	7.94E+00
$\ I - MA \ _2$	7.82E-09	3.64E-12	7.94E+00

Table 10 Computed relative error using exact inverse for various problems

Norms using exact inverse ($M = (A^T A)^{-1} A^T$)	<i>illc1850</i>	<i>well1033</i>	<i>small</i>
$\ AMA - A \ _2 / \ A \ _2$	4.99E-11	4.05E-13	6.01E-02
$\ MAM - M \ _2 / \ M \ _2$	8.94E-11	1.89E-12	9.62E-02
$\ (AM)^T - AM \ _2 / \ AM \ _2$	3.34E-11	6.33E-13	7.31E-02
$\ (MA)^T - MA \ _2 / \ MA \ _2$	7.82E-09	3.54E-12	9.86E-01

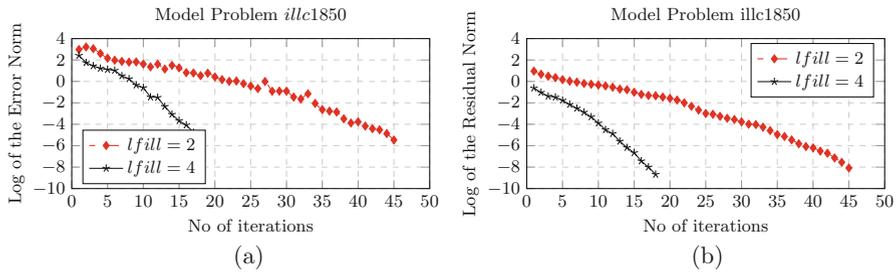


Fig. 1 Left:(a) Computed error estimates of the EPCGLS method, based on GASP scheme, with *lfill* = 2 and *lfill* = 4 for model problem *illc1850*. Right:(b) Computed residual estimates of the EPCGLS method, based on GASP scheme, with *lfill* = 2 and *lfill* = 4 for model problem *illc1850*

method, based on GASP scheme, with *lfill* = 2 and *lfill* = 4 for the model problems of Table 1 are given.

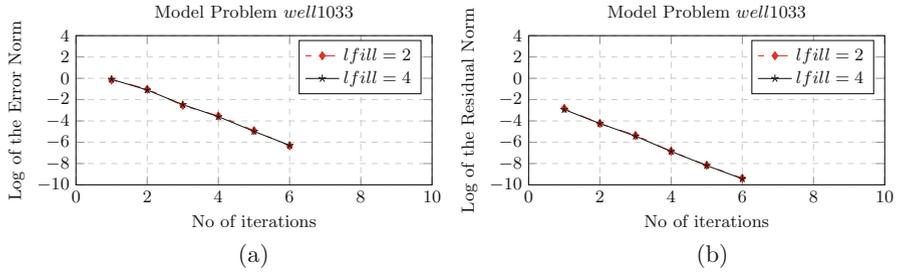


Fig. 2 Left:(a) Computed error estimates of the EPCGLS method, based on GASP scheme, with $lfill = 2$ and $lfill = 4$ for model problem *well1033*. Right:(b) Computed residual estimates of the EPCGLS method, based on GASP scheme, with $lfill = 2$ and $lfill = 4$ for model problem *well1033*

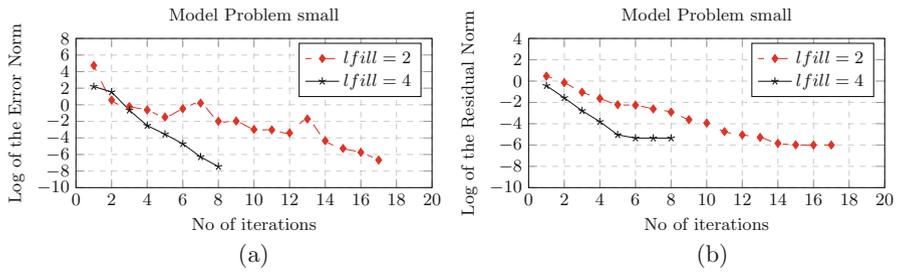


Fig. 3 Left:(a) Computed error estimates of the EPCGLS method, based on GASP scheme, with $lfill = 2$ and $lfill = 4$ for model problem *small*. Right:(b) Computed residual estimates of the EPCGLS method, based on GASP scheme, with $lfill = 2$ and $lfill = 4$ for model problem *small*

It should be noted that

- (a) The model problem *illc1850* requires 45 iterations with $lfill = 2$ and 18 iterations with $lfill = 4$.
- (b) The model problem *well1033* convergences in 7 iterations with either $lfill = 2$ or $lfill = 4$.
- (c) The model problem *small* requires 17 iterations with $lfill = 2$ and 8 iterations with $lfill = 4$, [26].

It should be stated that the convergence behavior of the EPCGLS method, based on GASP matrix, improves by increasing the value of $lfill$ parameter for solving large sparse least squares problems, [26]. Furthermore, the class of GASP matrices satisfies the modified Moore-Penrose conditions, and the numerical results are in qualitative agreement with the theoretical estimates.

6 Conclusion

The Explicit Preconditioned Conjugate Gradient Least Squares method, based on generic approximate sparse pseudoinverse (GASP) scheme for solving linear least squares problems has been presented. The GASP scheme is an approximate pseudoinverse matrix in conjunction with approximate pseudoinverse sparsity patterns, based on the modified incomplete QR factorization techniques. The sensitivity of the GASP matrix for solving linear least squares problems has been examined. The GASP matrix satisfies the modified Moore-Penrose conditions and the empirical results obtained were in qualitative agreement with the theoretical estimate derived. Furthermore, the given estimates can be used to assess the effectiveness of the method, taking into consideration the parameters, when computing the approximate pseudoinverse. Finally, the Explicit Preconditioned Conjugate Gradient Least Squares method, based on generic approximate sparse pseudoinverse (GASP) scheme, which possesses inherent parallelism, can be compared favorably to existing methods for solving a wide class of linear least squares problems.

Acknowledgments The authors acknowledge the Greek Research and Technology Network (GRNET) for the provision of the National HPC facility ARIS under project PR004033-ScaleSciCompII and PR006053-ScaleSciCompIII.

References

1. M. Arioli, I.S. Duff, Preconditioning linear least-squares problems by identifying a basis matrix. *SIAM J. Sci. Comput.* **37**(5), S544–S561 (2015)
2. M. Arioli, I.S. Duff, P.P. de Rijk, On the augmented system approach to sparse least-squares problems. *Numer. Math.* **55**(6), 667–684 (1989)
3. Z.Z. Bai, I.S. Duff, A.J. Wathen, A class of incomplete orthogonal factorization methods I: Methods and theories. *BIT Numer. Mathem.* **41**(1), 53–70 (2001)
4. M. Benzi, M. Tuma, A robust preconditioner with low memory requirements for large sparse least squares problems. *SIAM J. Sci. Comput.* **25**(2), 499–512 (2003)
5. A. Bjorck, Component-wise perturbation analysis and error bounds for linear least squares solutions. *BIT Numer. Math.* **31**(2), 237–244 (1991)
6. A. Bjorck, *Numerical Methods for Least Squares Problems* (SIAM, 1996)
7. A. Bjorck, T. Elfving, Accelerated projection methods for computing pseudoinverse solutions of systems of linear equations. *BIT Num. Math.* **19**(2), 145–163 (1979)
8. R. Bramley, A. Sameh, Row projection methods for large nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.* **13**(1), 168–193 (1992)
9. R. Bru, J. Marin, J. Mas, M. Tuma, Preconditioned iterative methods for solving linear least squares problems. *SIAM J. Sci. Comput.* **36**(4), A2002–A2022 (2014)
10. E. Chow, A priori sparsity patterns for parallel sparse approximate inverse preconditioners. *SIAM J. Sci. Comput.* **21**(5), 1804–1822 (2000)
11. T.A. Davis, Y. Hu, The University of Florida sparse matrix collection. *ACM Trans. Math. Softw.* **38**(1), 1–25 (2011)
12. C.K. Filelis-Papadopoulos, G.A. Gravvanis, Hybrid multilevel solution of sparse least-squares linear systems. *Eng. Comput.* **34**(8), 2752–2766 (2017)

13. C. Filelis-Papadopoulos, G.A. Gravvanis, E.A. Lipitakis, A note on the convergence rate of a class of approximate sparse inverse matrix methods, in *Proceedings of the 20th Pan-Hellenic Conference on Informatics*, Art. No 11 (ACM, 2016), pp. 1–6
14. G.H. Golub, C.F. Van Loan, *Matrix Computations*, 4th edn. (The Johns Hopkins University Press, Baltimore, 2013)
15. G.A. Gravvanis, The rate of convergence of explicit approximate inverse preconditioning. *Int. J. Comput. Math.* **60**(1-2), 77–89 (1996)
16. G.A. Gravvanis, Explicit approximate inverse preconditioning techniques. *Arch. Comput. Meth. Eng.* **9**(4), 371–402 (2002)
17. G.A. Gravvanis, High performance inverse preconditioning. *Arch. Comput. Meth. Eng.* **16**(1), 77–108 (2009)
18. G.A. Gravvanis, C.K. Filelis-Papadopoulos, E.A. Lipitakis, A note on the comparison of a class of preconditioned iterative methods, in *2012 16th Panhellenic Conference on Informatics (IEEE, 2012)*, pp. 204–210
19. G.A. Gravvanis, C. Filelis-Papadopoulos, E.A. Lipitakis, On numerical modeling performance of generalized preconditioned methods, in *Proceedings of the 6th Balkan Conference in Informatics (ACM, 2013)*, pp. 23–30
20. M. Hegland, On the computation of breeding values, in *CONPAR 90 – VAPP IV* (Springer, 1990), pp. 232–242
21. A.S. Householder, A class of methods for inverting matrices. *J. Soc. Ind. Appl. Math.* **6**(2), 189–195 (1958)
22. A. Jennings, M. Ajiz, Incomplete methods for solving $A^T Ax=b$. *SIAM J. Sci. Stat. Comput.* **5**(4), 978–987 (1984)
23. S. Kharchenko, L.Y. Kolotilina, A.A. Nikishin, A.Y. Yeregin, A robust AINV-type method for constructing sparse approximate inverse preconditioners in factored form. *Numer. Linear Algebra Appl.* **8**(3), 165–179 (2001)
24. N. Li, Y. Saad, MIQR, a multilevel incomplete QR preconditioner for large sparse least-squares problems. *SIAM J. on Matrix Analysis and Applications* **28**(2), 524–550 (2006)
25. A.D. Lipitakis, C.K. Filelis-Papadopoulos, G.A. Gravvanis, D. Anagnostopoulos, A note on parallel approximate pseudoinverse matrix techniques for solving linear least squares problems. *J. Comput. Sci.* **41**(101092) (2020)
26. A.D.E. Lipitakis, C.K. Filelis-Papadopoulos, G.A. Gravvanis, D. Anagnostopoulos, A class of generic approximate sparse pseudoinverse matrix technique based on incomplete QR factorization. submitted (2019)
27. A.T. Papadopoulos, I.S. Duff, A.J. Wathen, A class of incomplete orthogonal factorization methods. II: Implementation and results. *BIT Numer. Math.* **45**(1), 159–179 (2005)
28. R. Penrose, A generalized inverse for matrices, in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 51 (Cambridge University Press, 1955), pp. 406–413
29. J. Scott, M. Tuma, On positive semidefinite modification schemes for incomplete Cholesky factorization. *SIAM J. Sci. Comput.* **36**(2), A609–A633 (2014)
30. X. Wang, K.A. Gallivan, R. Bramley, CIMGS : An incomplete orthogonal factorization preconditioner. *SIAM J. Sci. Comput.* **18**(2), 516–536 (1997)

Undergraduate Research: Bladerunner



Adina Paddy, Cha Xiong, Colt Henderson, Tuu Le, and Daren Wilcox

1 Introduction

In this project, students have worked with Omni International to design and test a prototype 2nd Gen Prep96. This new automation platform will be built on the latest industry-standard electric drives and controllers. Omni has given the project the code name “Bladerunner.” Bladerunner will replicate the current X , Y , and Z Prep96 operation using a FESTO PLC and HMI. The X , Y , and Z axes are controlled by FESTO EXCM-30 H-gantry and an EGSC-BS-KF lift, respectively. The functionality and integration potential of the base model Prep96 must be preserved. This model will be programmed with CodeSys software and controlled by a PLC through an HMI interface. The current Prep96 uses a Windows PC with the user interface in C++.

2 Summary

The Prep96 is an automated homogenizer created, built, and sold by Omni International to life sciences laboratories. The Prep96 received its name from the machine’s ability to automatically homogenize up to 96 samples. In this project students established software communication between all products for programming tasks and future tasks. This project utilized a FESTO PLC and HMI with CodeSys

A. Paddy (✉) · C. Xiong · C. Henderson · T. Le · D. Wilcox
Department of Electrical Engineering Technology, Kennesaw State University, Marietta, GA,
USA
e-mail: apaddy@students.kennesaw.edu; Cxiong1@students.kennesaw.edu;
Chende46@students.kennesaw.edu; Tle62@students.kennesaw.edu; Dwilcox6@kennesaw.edu

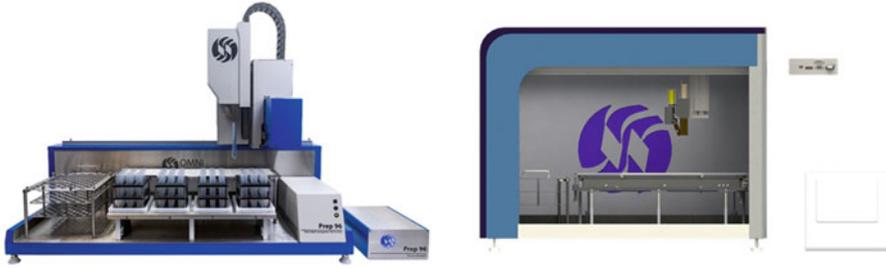


Fig. 1 Prep96 (left), Bladerunner (right)

programming to control Prep96 devices and replicate Prep96 x , y , and z motors with FESTO x , y , and z motors as well as all operations of the Prep Module. The Bladerunner will be more modular, cheaper to manufacture, capable of easy integration with other devices, and easier to service in the field (Fig. 1).

The Prep96 used microcontroller and C# based program to drive a servo motor and operate the machine. The disadvantage of Prep96 is that the program is hard coded; therefore, it is hard to modify. Mechanical limitations include an issue with position. A position error increases over time because the belt used to move the mixers will wear out, causing a slippage between the motor movement and the mixers. Electrical limitations include a combination of the complexity of the circuit boards and electrical components with the increase in the amount of time to produce a Prep96. The project Bladerunner uses FESTO PLC and motor controls aimed to mitigate these problems and increase the operating speed, as well as the manufacturing time. An advantage of the Bladerunner is that it uses a FESTO CDPX PLC with a built-in HMI, which increases the flexibility of the machine. The IEC 61131-3 language is easy to maintain and modify along with the touchscreen HMI and is easy to modify and adapt to a customer's inquiry. The FESTO H-gantry and motor controller with built-in soft motion control is a perfect solution to control its movement. The soft motion control includes real-time motioning; if any slipping happened on the operation, the motor control will automatically generate more train pulses to compensate for the slipping to drive the mixers to their designed position. The motor controller will reduce the position error rate at 1%, keeping machine running smoothly and continuously. Using the FESTO PLC will reduce the amount of electrical and mechanical component needed for the construction, which in turn will significantly reduce the size and weight of the Bladerunner.

3 Software

Two software applications are used in this project, CodeSys IDE and Designer Studio. CodeSys is the IDE used to program the PLC and the Designer Studio IDE is used to design the HMI (Fig. 2).

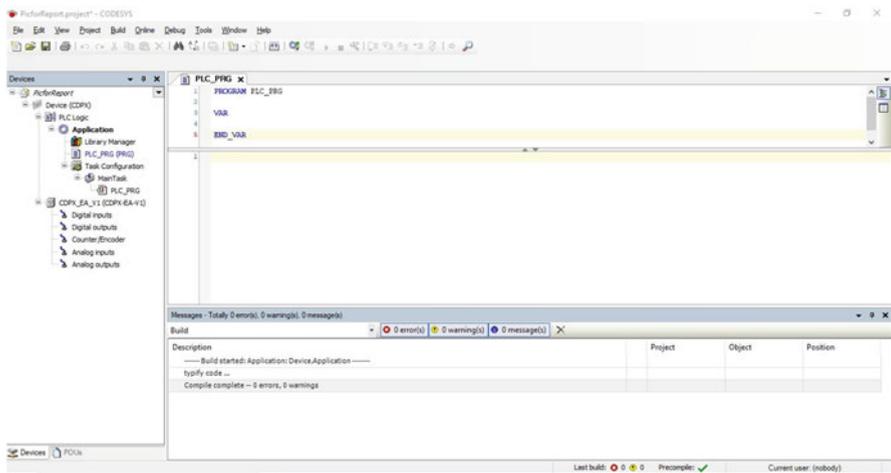


Fig. 2 CodeSys IDE

The CodeSys Development System is a IEC 61131-3 automation software developed and sold by the 3S-Smart Software Solutions GmbH. This project used CodeSys Development System V3.5 SP15 which allows programming in Ladder Logic, Structured Text, Function Blocks, and Instructions List formats simultaneously. Variables can be created, initialized, and given tag locations in the GVL or Global Variable List and can later be called within different languages. On the left side of the Workspace is the Device window; the CDPX and the CDPX-EA-V1 is added to the program and all initial PLC settings are adjusted. Under Application is PLC_RPG where the PLC program is written; multiple programs can be added. The PLC_RPG has two windows. The top window can be used to initialize the local variables and the bottom window is used to write code. A Designer Studio is an application with a drag-and-drop interface used to create graphical HMI pages. The software comes with two different programs: the HMI Client and the Designer Studio itself. The HMI Client is a program that can remotely view and control an HMI given its IP address. In the left window under it is the Project View that presents the project files in the form of a hierarchical Project Tree. On the right side is the Properties of a selected item in the Working Area on the center of the page. Also, on the right in a tab is the Widget Gallery. This gallery consists of preprogrammed buttons, shapes, text fields, etc. that were used in this project. Tags are exported in an .xml file from CodeSys and imported into the Designer Studio to be addressed with input or output widgets (Fig. 3).

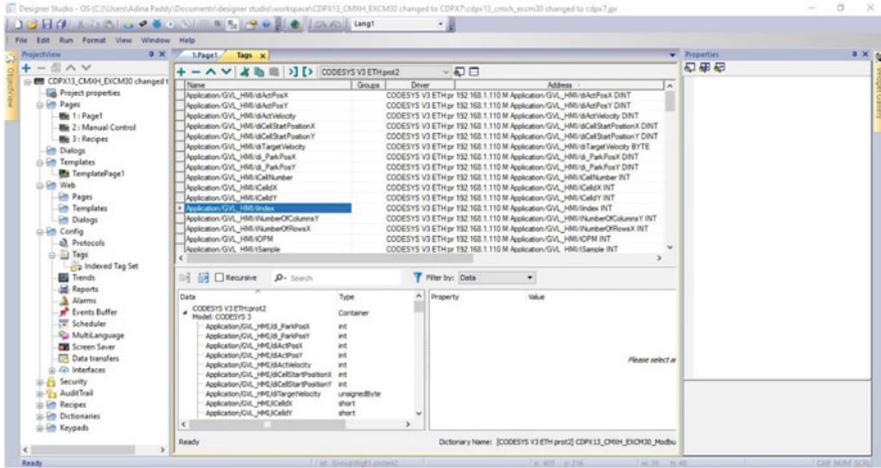


Fig. 3 Tags Import Page

Table 1 Proposed checklist table

Proposed	Status
Visual indication of sensor detection on a Boolean variable	Completed
Control of solenoid with a button	Completed
Control of motor enable with a button	Completed
Control of analog speed with a slider and numeric user input	Completed
Can jog on X and Y axes	Completed
Can change location from user input	Completed
Can move in a sequence based on user numeric input on H-gantry	Completed
Z-axis control	On Hold
Pick and place probes with CodeSys programming and interface through HMI	Partially Completed
Overall Prep96 operation sequence programming	On Hold

4 Results (Table 1)

4.1 Programming Results

Procedure The Bladerunner program is designed to perform the following sequence accurately and orderly. The procedure program Bladerunner uses a Sequential Function Chart (SFC) which is a combination of Ladder Logic and Structure Text. The advantage of SFC is it allows the user to split the whole procedure of the program into many steps that can run independently.

The Global Variable List or GVL is where variables can be named, declared, and given an address if needed. Variables can be declared variables with varying data types such as an int, an array, or a boolean. Addressing the variable is giving

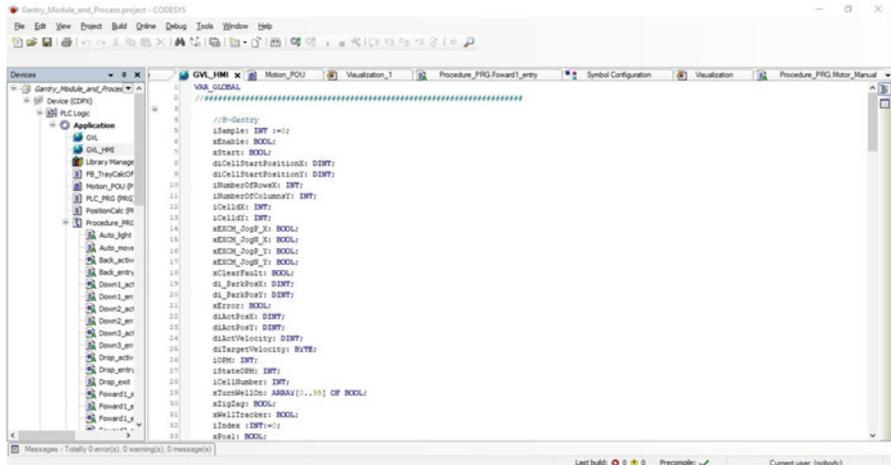


Fig. 4 GVL_HMI

it a physical digital/analog port. With these ports, you can send signals or receive signals to and from the PLC. Doing so is giving the variable a “Tag.” Variables can be declared in the main program, but such variables would be private to that individual program. The GVL allows any PRG to utilize the list of variables within the application. This can allow you to use variables in one code, jump to another code and do calculations there, and then return to the main code with the values intact.

Figure 4 shows the main components in an SFC program (Fig. 5):

1. **Transition stage.**
2. **Step:** Every step contains three substeps.
3. **Step_entry (optional):** The program in Step_entry will run one time before the step is active.
4. **Step_active:** The main program in the step. The program in Step_active will keep running and monitoring repeatedly. The Step_active finishes when the transition stage for the next step becomes TRUE.
5. **Step_exit (optional):** When the transition stage for the next step becomes TRUE, the program in the Step_exit will be run one time before it exits and moves to the next step.
6. **The subprogram called action written for Step_entry, Step_active, and Step_exit.** This subprogram can be written in different programming language as LD, ST, or FBD.
7. **Associated action (optional):** the associated action will run parallel and repeatedly with Step_active.
8. **Jump:** Jump command helps change the direction of the program to pointed step.

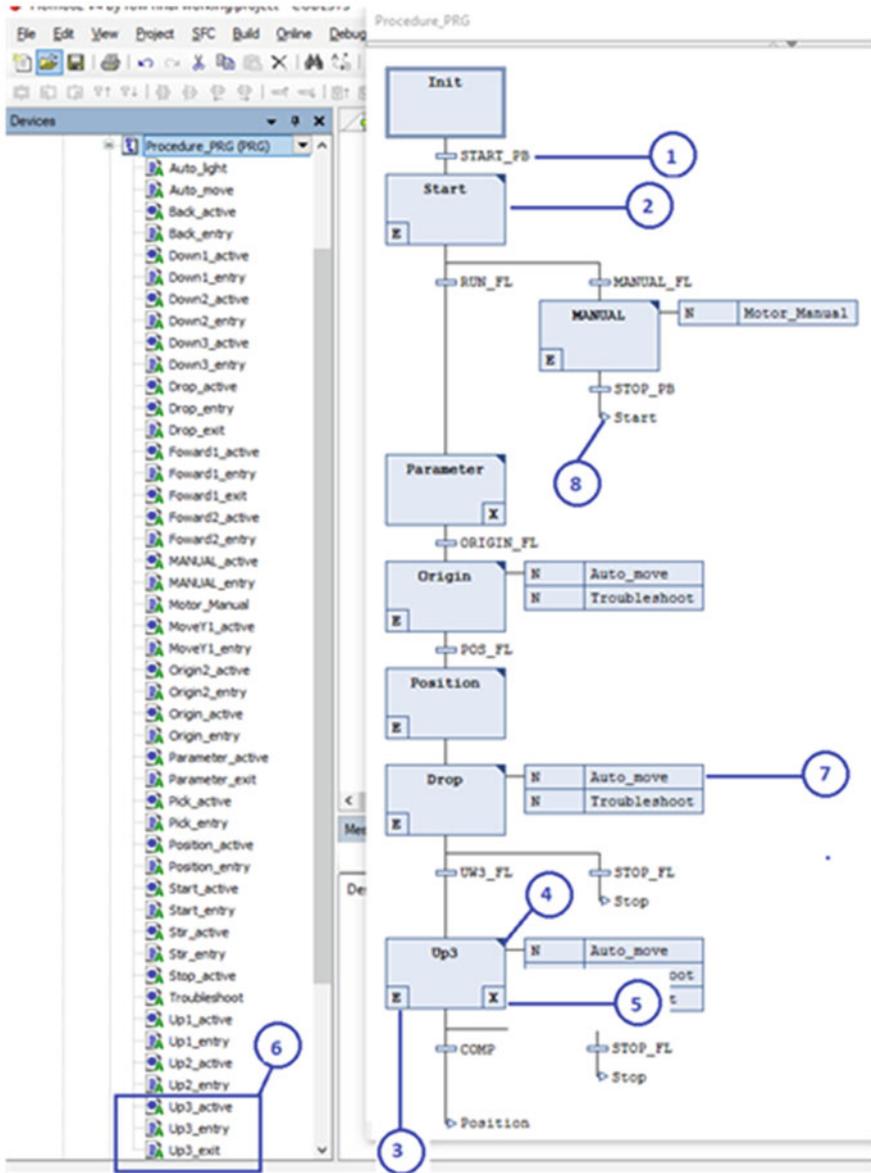
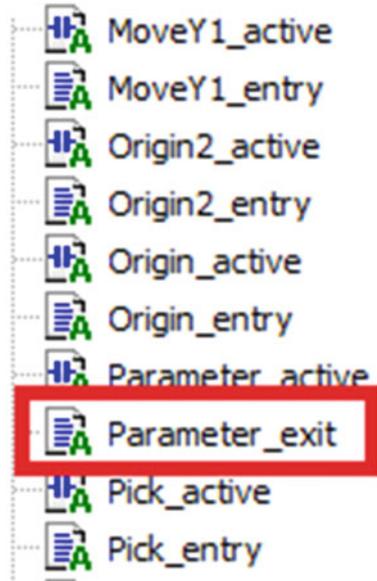


Fig. 5 SFC flowchart

The Bladerunner program is designed to perform the following sequence accurately and orderly.

Fig. 6 Parameter exit code



Start → Input Parameters → Move to default set position → Move forward to first Probe position → Move forward second time → Stir → Move Back → Move Down → Drop Probe.

This procedure will work with samples in order by row. After doing the first turn, the flow will jump to the Position step to check if the current row is equal to the number of rows in the input tray. If the current row is not equal to the last row, then move the mixing block to a new row in y direction with a distance equal the length of the mixing block. If the current row equals the last row in the tray, move and then set the position origin in y direction to calibrate. Next, move to the next column in the x direction. If the completed sample equals to the input sample, the program will go into Stop stage. To start again, press the Reset button and the Start button (Fig. 6).

The variables used in the program are declared in GVL and GVL_HMI. The setup for tray dimension, Stir time, Stir Velocity, and Moving Distance can be modified and updated in the file named parameter_exit. The programs named Auto_move and Auto_light are used to move siders and light up array bulbs used for the purpose of testing and demonstrating the code only.

After pressing Start, the program goes into Parameter Step waiting for user inputs. Inputs include Input Sample, Tray Type, String Velocity, and Stirring Time, then the OK button can be pressed. The program will go into the parameter_exit action to process one input data before going to the Origin 1 program to move the mixing motor to default position.

```

38
39 // ***** Tray type 3 *****
40 ELSIF (TRAY_TYPE = 3) THEN
41     TRAY_ROW := 12;
42     TRAY_COL := 8;
43     x1_dist := 10; // x distance delta_x between 2 columns
44     y1_dist := 10; // y distance delta_y between 2 rows
45
46
47     f1_dist := x1_dist;
48     f2_dist := 40; // fix distance between new stick and new sample on x directio
49     u1_dist := 50; // up distance z after pick a new stick
50     u2_dist := 50; // up distance z after stirring
51     u3_dist := 50; // up distance z after drop used stick
52     d1_dist := 50; // down distance z to pick a new stick
53     d2_dist := 50; // down distance z to begin stirring
54     d3_dist := 50; // down distance z to drop used stick
55 END_IF
56
57 orgx := 10;
58 orgy := 0;
59 orgz := 0;
60 NOZZEL:=4; // Number of nozzels
61 block := (NOZZEL-1)* y1_dist; // dimation of block of 4 nozzels
62 b1_dist :=f1_dist + f2_dist ; // back distance x to drop used stick
63
64 // ***** Profile of trays ends here. *****
65
66 // ***** Stiring time input. *****
67
68 INPUT_TIME := INPUT_TIME*1; // * 1000 input = second ; * 1 input = millisecond
69 STIR_TIME := TO_TIME(INPUT_TIME);
70
71 // ***** Stiring Velocity input. *****
72 IF (STIR_VELOCITY > 5000) THEN
73     STIR_VELOCITY:= 5000;
74 END_IF
75
76 speedMotor1 := TO_DINT(STIR_VELOCITY);
77 //speedMotor1 := SpeedMotor2;

```

Fig. 7 Tray profile in parameter_exit

In parameter active action, the program will keep running while waiting until the OK button is pressed to set the bit ORIGIN_FL (origin flag) = 1. When the ORIGIN_FL is set, the program will leave the Parameter Step and go to Origin. Before leaving the Parameter Step, the parameter_exit action is run one time (Fig. 7).

Depending on the tray type selected, the tray input data to operate the homogenizer machine will change.

- Tray_Row: Defined number of rows in the tray
- Tray_Col: Defined number of columns in the tray
- X1_dist: Distance delta_x between two columns
- Y1_dist: Distance delta_y between two columns
- F1: Distance to move forward is to pick up new probe in x direction
- F2: Distance between the first column of the probe tray to the first column of the sample tray in x direction
- U1, u2, u3: Distance to move up in z direction
- D1, d2, d3: Distance to move down in z direction
- B1_dist: Distance to move backward to drop the probe in x direction
- Nozzel: Number of mixing heads



Fig. 8 Set and Reset flag

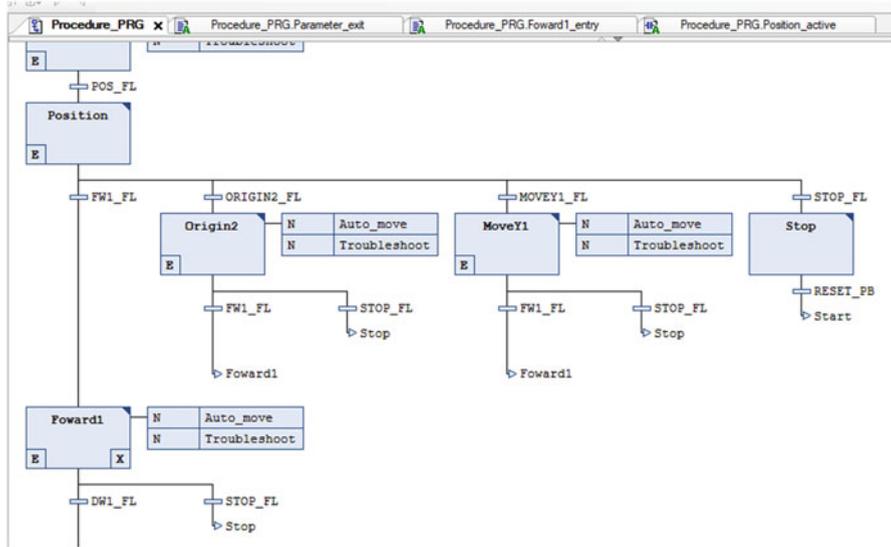


Fig. 9 Position and Forward 1

- Block: The length of one set of nozzles
- Orgx, orgy, orgz: Defined three setup origins for directions x, y, and z
- Stir_time: Time for stirring in ms
- Stir_velocity: Stirring velocity in kRPM

After finishing a cycle, the program will jump to the Position step to select the next route of code for the new operating cycle. In the Position active program if FW1_FL is set, the program will go to Step Forward 1 and will move forward to pick up a new probe. If MOVEY1 is set, the program will move to the MOVEY1 code block and to a new row in y direction. If STOP_FL is set, the whole process is completed, and the code will wait for the Reset button to start a new turn. When the flag moves, the next step is set, and the flag for the current step should be reset (Figs. 8 and 9).

In the entry action, depending on whether the flag is set and on the current conditions, the distance to move the mixing head to the next destination position is loaded to the variable movestep (for moving x direction) or moveystep (for moving y direction) or movezstep (for moving z direction) before the motors move. The variables istep, iystep, and izstep respond to the real-time values of the x, y, and z axis of the mixing head (Fig. 10).

```
Procedure_PRG.Foward1_entry x
1 IF FW1_FL AND (COMP_TURN = 0) AND (ROW_POLS <= 1) THEN
2 //istep :=0; movstep := fl_dist ; if using relative coordinate
3
4 ELSIF FW1_FL AND (COMP_TURN <>0) AND (ROW_POLS = 1) THEN
5 movstep := istep + (2*fl_dist) ;
6
7 ELSIF FW1_FL AND (COMP_TURN <>0) AND (ROW_POLS > 1) THEN
8 movstep := istep + fl_dist ;
9
10 END_IF
```

Fig. 10 Forward1_Entry



Fig. 11 Forward1_active

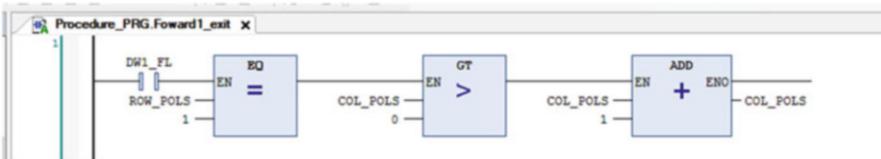


Fig. 12 Forward1_exit

In active action, the enable bit is turned on to make the motor move in the desired direction. Keep comparing the values in istep and movstep. If the value of istep >= movstep turns off the enable bit to stop the motor, then reset the FW1_FL and set DW1_FL to do the next step. All other moving steps behave similarly (Fig. 11).

Before moving to the next step, users must go into the exit step to run the code one time and update the current column to the variable COL_POS (Column Position) (Fig. 12).

In the Pick_entry action, the output of the solenoid is set to 1 (active) to pick up the probe, and the code in the Pick_active action will check if the input from the sensor does not set to 1 (see the probe). If the solenoid is 1 and the sensor is 1 at the incorrect time, the Error flag is set, and the error program will run and pause the machine. The error light will be solid on and the stop light will flash. In the Drop_entry action, the output of the solenoid is set to 0 (deactive) to drop the probe,



Fig. 13 Troubleshoot action

and the code in Drop_active action checks if the input from sensor does not set to 0. If the probe is still stuck on the solenoid, the Error flag will be set and the error program will run to pause the machine, and the error light will turn on and the stop light will flash.

The Stir motor is controlled by one analog output and digital output on the PLC depending on the velocity (in kRPM) and the time set in the Parameter Step. The velocity value is calculated and converted to mV (in parameter_exit action). The value is sent to the analog output on the PLC to the control speed and run time for the stirring motor. In Stir step, the Stir_entry the bit will reset the value of the stirring timer and set the enable bit of the stirring motor. In Stir_active the bit will wait for the output of the stirring timer (TON) to be 1 then reset the bit of motor and STIR_FL then set UW2_FL to go to the next step.

The Pause, Stop, and Error codes are all in block named Troubleshoot. Troubleshoot is always running since it is associated in every step. The input of the Stop button is also linked to an input of an up counter. When the Stop button is pressed one time or the ERROR_FL is set, the RUN_FL (Run Flag) will be reset. This will make the current operation stop or pause, the Run Light will turn off, and the Stop light starts flashing. If the Start button is pressed, the counter will reset to 0, the RUN_FL will set the program to continue to run, and the Run Light will be on. In this case, the system is paused by the Error_FL set (Fig. 13).

After troubleshooting the errors, the user needs to press Reset and then Start to run the machine. If the Stop button is pressed two times, the Stop_FL is set. The program will then jump to the Stop stage and reset the run flag to pause the machine, the Run Light will be off, and the Stop Light will be solid on. Then the code will wait for the Reset button to be pressed to exit the Stop stage and jump into the Start stage with all current information reset. In the Start stage, the program waits for the Start button to be pressed to set the RUN_FL and input new parameters for a new process.

Gantry FESTO included libraries that have functions that make the coding easy. These functions include homing, movement, and feedback. Using these functions is

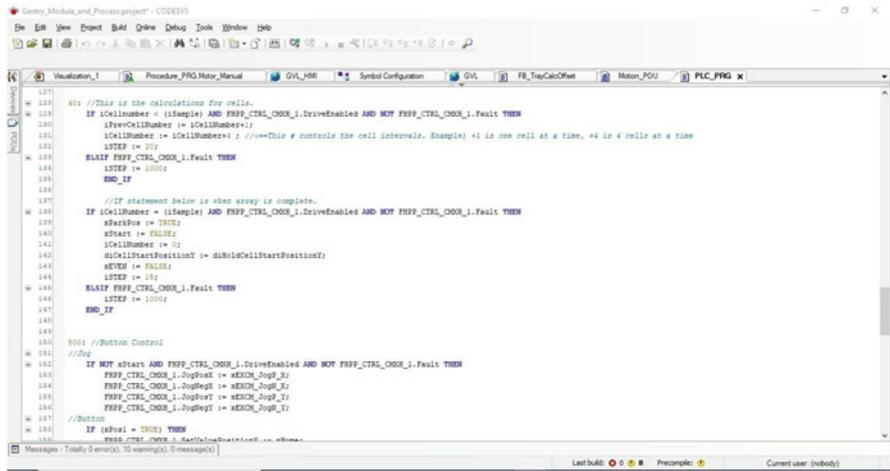


Fig. 14 Cell and jog sequence

as simple as setting a bit or clearing a bit. Using the libraries that FESTO provides, the code will move the gantry to certain locations. Depending on the button that is pressed, the code will execute the code for the button pressed. When the main program starts, the code will repeat on loop until the desired number of tubes has been completed.

The library is compiled with variables that either respond to true or false or output a value. Using the library is as simple as calling the library’s name and then adding a “<variable name>” to the end of the statement. With the variables, we can also use conditional statements to check if one variable has been triggered or not. This can allow the code to check itself to make sure that something is not wrong. When something is wrong, the code will set the “Fault” variable allowing the user to see that something is wrong.

The gantry control is programmed using the Structure Text with the CASE instruction. CASE allows several conditional instructions containing the same condition variable into a construct. For example, if the value of the variable <Var1> is <value i>, then the instruction <instruction i> is executed. This will allow the code to execute all the codes within the case statement and, then when it is finished, compare the variable with another case. The program first finds the case statement it is going to and then execute the commands in that case. When approaching the last instruction in the case statement, the code reassigns a new value for the case variable, allowing the code to go to a new case statement after it completes its task (Fig. 14).

Module The manual DIO controls are programmed in Structured Text, shown below; the program is straightforward and basic, utilizing IF statements. The IF statement is used for checking a condition and, depending on this condition, for

executing the subsequent statements. A condition is coded as an expression that returns a Boolean value. This code can be modified to include control for up to three other modules for a total of four.

The first IF statement here in the example will return all values back to the original state. The values are restored back to the original state to make sure they are not off. The code then determines how fast the motors should be spinning. The motor controller is a 5 V analog, so the code sends voltage based on the user's settings. The user can then input a number to use or use the slider to set the speed. On the HMI, the user can push a button that can trigger the solenoid. The solenoid is linked to the Soleactive variable. Depending on this value, it can set/trigger or clear/release the solenoid.

5 Conclusion

Development of Bladerunner to date was accomplished with demo equipment provided by Omni and Festo. Funds have not been released by Omni for the purchase of the z axis equipment which has been placed on hold. The basis of all other features has been delivered to Omni, namely: visual indication of sensor detection on a Boolean variable, control of solenoid with a button, control of motor enable with a button, control of analog speed with a slider and numeric user input, and movement in sequence based on user numeric input on H-gantry. When additional funds are released in the new fiscal year, Omni engineers will implement the z axis, customize the software in a proprietary sequence, and complete the cabinet design. Bladerunner will transition into the prototyping phase before production.

Comparison of the IaaS Security Available from the Top Three Cloud Providers



L. Kate Tomchik

1 Introduction

Cloud computing servers are a black box to the purchaser – a computer system built and maintained by the vendor. The stress of sustaining the environment has been removed from the customer, allowing them to now focus on the software development. Even in the cloud, however, the customer must understand how the system is secured from malicious attacks. It may be prudent to open the black box and take a look inside to understand what is secured by the provider at the creation of a cloud server. A comparison of available online technical documentation detailing the security of the cloud providers will be compared for a unique perspective not provided in other research papers. In conclusion, recommendations for the best way to use and support cloud environments will be proposed. The main research contributions for this paper are from a variety of documents published over the last few years detailing what security needs to be established for any computer environment, as well as documentation provided from the top three vendors to understand the special needs for an off premise cloud environment.

This paper is structured to first discuss relevant research in the area of cloud security needs. The background section will describe the security threats and controls of all computer environments. Identifying which security threats are the responsibility of the cloud provider is accomplished by determining if those threats affect the virtualization, server, storage, or networking application layers. The final background topic will be a discussion of security requirements unique to the cloud.

The problem statement and discussion will cover three areas of concern for customers using cloud environments. These include the new agile programming

L. Kate Tomchik (✉)

Regular Research Paper Submission, Kennesaw State University, Marietta, GA, USA

e-mail: ltomchik@students.kennesaw.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_23

307

methodology, the need for better security training of employees, and the necessity of a cybersecurity specialist at every company. The evaluation section will provide original research comparing the security provided by the top three cloud platform vendors – Amazon, Azure, and Google – which has been taken from each cloud provider’s online documentation. Finally, recommendations to companies planning to use cloud resources will be presented, as well as the website for the Center for Internet Security, which provides documentation illustrating how to implement security on each of the three discussed cloud providers.

2 Related Work

Many papers provide essential background information related to the topic of cloud security which help direct the discussion in this paper. These papers also indicate a need to revisit this topic every year, as the security threats and controls are changing rapidly year after year as the use and accessibility of cloud platforms grow.

“Secured Cloud for Enterprise Computing” [1] discusses the four essential parameters of cloud computer trust: availability, reliability, turnaround efficiency, and data integrity. The cost of cybercrime will grow to 6 trillion dollars by 2021, while 97% of organizations will be using cloud environments. This paper explains a variety of security vulnerabilities, yet many of these apply to any computing environment and are not unique to the cloud. The paper missed an opportunity to explain what security is available to the customer from the cloud vendors to remediate the issues that are at the cloud platform layer. Regardless of whether a customer’s application is on physical servers or cloud IaaS servers, the customer must control and implement the application security access and use.

“Cloud-Trust – A Security Assessment Model for Infrastructure as a Service (IaaS) Clouds” [2] gives an extensive outline of many security threats to IaaS and how to protect from them in order to provide cloud trust. A large part of this discussion is on the frailties of the virtual machine (VM)/container/hypervisor usage of the cloud servers. The use of a VM allows the provider to share large computer clusters with multiple customers, but that creates a new threat over traditional use of physical servers. A demonstration of how to assess the security controls of a cloud service is presented, with discussion of the common vulnerability scoring system. The paper leaves the reader curious as to how the different cloud providers already provide tools to remediate these threats.

“A Security Proxy to Cloud Storage Backends Based on an Efficient Wildcard Searchable Encryption” [3] discusses a security proxy that can be used both on premises and in the cloud. Using a secure index with searchable encryption allows the cloud data to be quickly accessed while still guarded. This removes the concern of the cloud providing confidentiality and leaves that responsibility to the customer. The complex nature of this solution clearly shows that the implementation of data encryption is a detail best managed by an Information Technology Security Officer. It also supports the suggestion that many application developers will skip

the security solutions, at least initially, when they build test environments at a cloud provider.

“Cloud Data Auditing Techniques with a Focus on Privacy and Security” [4] explains three cryptographic methods for auditing: message authentication, homomorphic linear authenticaters, and Boneh–Lynn–Shacham (BLS)-based homomorphic methods. These methods can be used to secure audit data from the cloud to the third-party auditor (TPA). After reading this paper, it becomes obvious that to give the most secure level of auditing to a cloud environment, the auditing is ideally performed by a TPA or by a company’s internal security officer using auditing software that is located external to the cloud environment. Since the auditing will not be done on the cloud IaaS computer, the audit logging capabilities of each provider will not be part of this paper’s comparisons. “[T]he technical challenges of auditing services can be addressed by employing a separate architecture for auditing purposes.”

This paper will review the main security threats to any computer environment and define which application layer should be analyzed to limit that threat. Once the layer is determined, the threats that are controlled by the cloud provider of the Infrastructure as a Service (IaaS) computer are made clear. In addition, unique security concerns for cloud servers are explained. Once the list of IaaS security issues is identified, they will be systematically reviewed for each of the top three cloud providers – Amazon, Azure, and Google. In conclusion, recommendations for companies using cloud environments will be suggested.

3 Background on Security Threats

The Cloud Security Alliance (CSA) contains 98 different cloud security controls – all of which were previously defined for non-cloud security frameworks. This is because the cloud is simply a merging of many existing computer-related topics. Since areas like networking, web access, and virtualization already have defined security controls, it makes sense to use these same controls with the cloud operations [5].

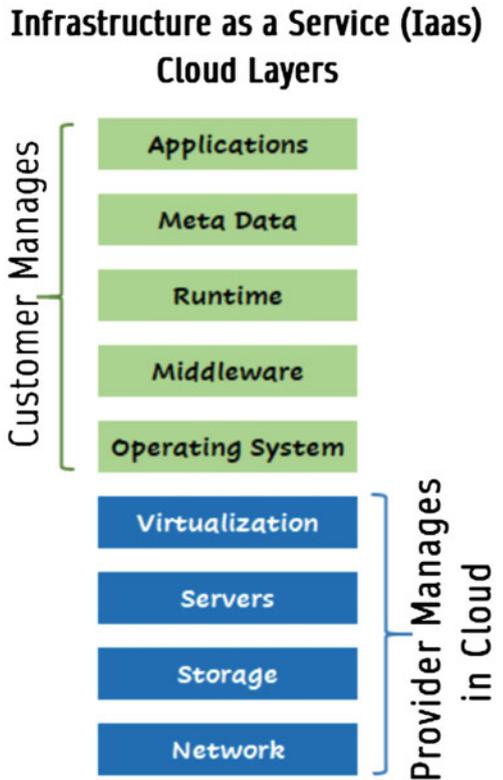
The primary categories of security risks are cross-referenced with the application layer of the Infrastructure as a Service (IaaS) platform. The cross-reference identifies which risks are to be remediated by the cloud provider and which risks will stay in the control of the IaaS customer. Ten categories of security threats and best practices for controlling those threats are in Table 1. The responsible party for managing each threat is dependent on which application layer can detect that threat. Some application layers are managed by the cloud vendor, while the remaining are still secured by the customer (Fig. 1) [6].

Unknown devices – A thorough inventory of all trusted devices must be maintained at a single location that is accessible when a new device attempts connection. Devices not on the approved list are then denied access to the cloud system until further research occurs. Hardware configuration standards must be

Table 1 “Building a Healthy Fleet” *BeyondCorp*

#	Threat	Control
1	Unknown devices	Inventory and asset management
2	Platform compromise	OS and base software configuration management
3	Security control bypass	Security police management and enforcement
4	Privilege escalation	Resilience against system takeover and persistence
5	Software compromise	Software control and anti-malware
6	Attack persistence	Remotely verifiable platform state
7	Authentication bypass	Robust authentication of platform and user
8	Data compromise	Data protection
9	Attack concealment	Logging and log collection for detection capability
10	Attack repudiation	Response capability on platform/ detection and response

Fig. 1 Cloud Infrastructure as a Service application layers with security responsibility identified



enforced so that security evaluations of the computers are uniform. This security is generally at the network application level, so it must be provided to the cloud provider of an IaaS server with input of the known devices provided by the cloud customer.

Platform compromise – Platform compromises occur when a system has an operating system or other software that has not been vetted by the organization. Strict lists of all software, including permissible versions of the software, must be maintained. Patching of software to a particular level must be regulated and recorded at a centralized location. Companies need to maintain a cadence to identify any software that has not been updated in the required duration since release. All allowed software with approved patch releases is to be maintained in a central location that can be accessed as needed. This is part of the approved software choices that will be made by a company's cybersecurity team for an IaaS environment.

Security control bypass – To bypass the computer's security controls, the cybercriminals attempt to avoid firewalls, intrusion detection/prevention code, and the anti-malware. Ensure the security controls are not bypassed by maintaining a complete list of approved programs for every server environment. Restrict all users from logging in as superuser, for example, "root" on a Linux server, so the individual using the account can be audited properly. Maintain the list of all authorized customer log-ins of the cloud IaaS environment, and ensure more than one individual can control the list.

Privilege escalation – Privilege escalation occurs when a cybercriminal takes advantage of a configuration oversight. Malware that employs keystroke logging or tracking cookies can be used to steal passwords and facilitate future privilege escalation attacks. In case this type of malware attacks a server, ensure the system has multiple layers of defense. The goal is to identify the virus before it is able to stop the server's audit logging capabilities. This is part of the software application layer and therefore the responsibility of the cloud customer.

Software compromise – Have controls in place to verify each program has been authorized before it is run on a server. An allowed list of programs for the server must be maintained and verified. Establish an approved program list in a central retrievable location that is updated as part of application installs to the IaaS environment. Insure that ineligible software trying to run on the cloud server is refused. The software application layer should be secured by the business customer.

Attack persistence – A persistence threat is an attack in which an unauthorized cybercriminal gains entrance to a computer network and rests there dormant. This allows the hacker possible access to restricted information and can lead to data theft. To address persistence threats, one can encrypt servers and provide a mechanism to ensure secure boot up, including stringent approval of remote users trying to connect. This is at the server level and therefore needs to be part of the cloud provider's hardware/VM installation, with alerting to inform the customer of attacks and remediation.

Authentication bypass – Today, it is common not only to request the user have a known user identification (ID) and a password but also to confirm that person with a hardware-isolated system – such as confirming with a code sent to a

cellphone. Bypasses could occur from weak tokens, unprepared data statements, unprotected data files, or application calls without authentication. Many developers fail to test the system security prior to the production release (<https://cyware.com/news/authentication-bypass-vulnerability-what-is-it-and-how-to-stay-protected-ccc2ea38>, accessed 3-8-2020). This is a security measure enforced at the virtualization level of the IaaS environment, so it must be available from the cloud provider with the authentication methods being provided by the customer.

Data compromise – Always assume that some of a server’s data will contain sensitive information. Data is often transported between systems and must be encrypted during the transport. The data at rest must be encrypted as well, so a stolen physical server cannot be read. It is important to have the ability to remotely erase a compromised device. Data can be compromised at both the storage and the network layers of the IaaS model. Some articles describe this as a customer requirement, but the cloud provider also needs to have these encryption tools available for implementation. Requiring the customer to add these measures independently increases the security risk of the purchased IaaS environment.

Attack concealment – Concealing an attack could leave a server at risk. Systems must assume that some breaches will occur, and if a threat is detected, it needs to be streamed to the external cloud auditor environment. Ideally, it is best to protect the audit logs from tampering and have them regularly reviewed by the security auditing team. The owner of the environment needs to be notified and updated on the nature of an attack. The logged data needs to be detailed with all dates, times, number of attack attempts, source of the attack, environment details of the attacker, changes made to the environment, behavior of the environment, and any other data that provides a complete picture of the attack. The cloud provider needs to audit the IaaS host at both the server and virtualization layers. Timely notification of suspicious log events is critical. Attack concealment can be performed by the cloud provider, but then there are no checks and balances of the data that results. The auditing of the activities on a server ideally is performed in a separate server using a third-party auditor (TPA).

Attack repudiation – A repudiation attack occurs when an application or server does not provide the ability to log the actions of the users. This allows a hacker to flood the server with connection requests or in other ways deny the customer authorized access to the server. Cloud users must rely on the ability to access servers at all times. The authorized security auditing team needs to create the timeline and incident report for proper research of any repudiation events. Policing of the server and virtualization layers are requirements of the cloud provider.

In an Infrastructure as a Service (IaaS) model, the primary security measures that must be supplied by the cloud provider are those that involve the networking, storage, server, and virtualization layers. The software tools and application programs are provided by the customer of the cloud provider [1].

Cloud platforms have special security concerns due to the fact that the cloud consumer expects to be able to expand the cloud resources quickly and efficiently while having these extensions immediately supported in the case of potential security holes that are added by the change. The need for availability would be

compromised by any type of denial of service (DoS) attack, an attack where the perpetrator tries to flood the connection ports of a host, forcing legitimate connections to time-out. These attacks are prevented by ensuring there is no application software compromise, by denying any unapproved program from access to the server.

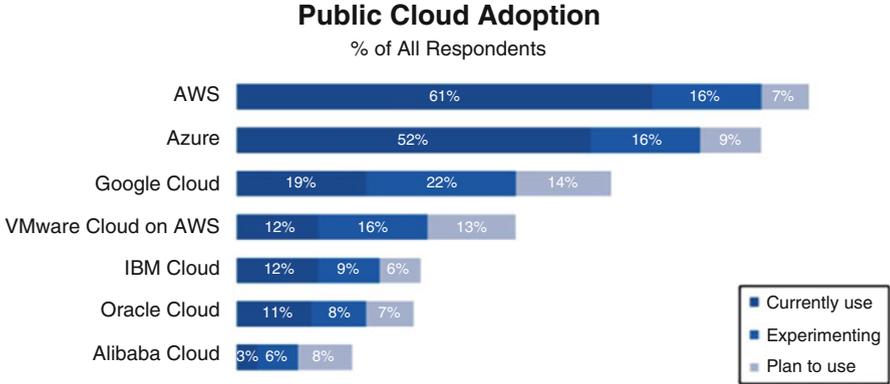
A second cloud-specific problem is the result of resource sharing with other cloud consumers. A breach of a resource will logically affect all the consumers sharing that resource. The customer needs to know that their data will be encrypted on the IaaS server. This protects the data from being viewed by unauthorized users. In addition, encryption in transit must be utilized during initial setup and for any continued connections through the applications or for replication. Resource-sharing is often listed as the primary reason customers will resist moving applications to the cloud, as adding this securely is a complex operation.

Cloud providers attempt to virtualize each server to separate them from other resource-sharing customers. This virtualization or containerization must include all the security measures needed to completely separate one consumer from another. The toughest cloud security provision is that of availability – cloud providers must quantify how much and when each consumer requires the resources. This is not just to charge the customer fairly but also to ensure all the consumers have access to their applications and the resources needed to run their operations, at least to the agreed-upon performance parameters paid for by the end user.

In non-US locales, data locality is a major concern. The fear is that their data may end up being housed in the United States on US-based servers, which could expose their businesses to spying by the US intelligence community. This concern can only be allayed if multiple non-US locations of data centers are available. Data locality has become a considerable issue for non-US customers [7].

The security provided by the top three cloud providers, which, as of August 15, 2019, include Amazon, Azure, and Google, will now be compared based on the information contained in their respective websites and service documentation (Fig. 2) [8]. Due to a recent win of a major 10 billion dollar Pentagon contract, it is likely that Azure will become the most widely used cloud provider in the next couple of years. The pentagon was criticized for only using one cloud provider, as that in itself can pose a security risk should Azure not be able to handle the increase demand from the contract. Interestingly, Google withdrew from the contract bidding, citing an ethical issue with using artificial intelligence to man weapons [9]. So the top three may rotate positions, but it is still likely to be the same three players in 2020.

These three providers have been described as the leaders for cloud Infrastructure as a Service. A leader is one that has the ability to execute and has a completeness of vision [10].



Source: RightScale 2019 State of the Cloud Report from Flexera

Fig. 2 Rightscale 2019 state of the loud report from Flexera

4 Problem Statement and Discussion

Each of the top cloud providers has unique ways of mitigating security issues, and each provides methods to secure the Infrastructure as a Service (IaaS) platform. A few trends for cloud development which can cause problems in the area of IaaS security will be presented to clarify the evaluation of these security options.

4.1 Problem 1: Agile Programming

A new cloud Infrastructure as a Service (IaaS) server takes less time to initiate than it takes to drive to Micro Center to buy a new laptop. The burden of maintaining the equipment, setting up the network, and providing the storage has been removed from the customer. Even the increase of CPU and memory capacity during peak usage periods is easily accomplished, and they backed out when no longer needed. For these reasons, application developers are now using cloud resources for a quick route to build a new system. These servers no longer have the hardware teams ordering from the approved vendors or the security administrators verifying the software chosen and overseeing the installation the way they did for the physical computer purchases. The responsibility of securing the virtualization, server, storage, and networking application layers is now that of the cloud provider.

The need for a developer to quickly provision a server is a consequence of the popularity of agile programming methodology. The days of waiting for computers to be ordered, purchased, installed with software, and even secured are gone when agile developers can instead reach out to the clouds. “Agile development in cloud computing environment is an important area in software engineering” [11]. The

largest challenge posed by this new relationship between agile developers and the cloud was reported by 18% of customers surveyed to be that of security and privacy.

The initial build of a new cloud platform in itself presents a security concern. The act removes some of the data protected by a computer in a secured room to somewhere not as tactile. Most users will want to seed some of the data from their corporate location. This data must be secured whenever the new cloud server sends or receives traffic. This is why it is so critical that the security of cloud providers be provided “off the shelf” and not require complex definition and setup.

4.2 Problem 2: Security Training for Employees

Security training and employee monitoring are a requirement for every business. “Today’s data breaches often seem to be caused not just by malware infections or external threat actors, but human error, insiders with an ax to grind, and simple security failures” [12]. No matter how much security is provided by a cloud provider, if the end user does not know how to implement it or, worse, chooses to circumvent it, there is a security risk. It is important to invest in the employees to help properly train them in the security needs of the corporation. This decision needs to be seen as a priority at the board of director level to be successful. “Gaining management buy-in to fund and encourage security awareness training will be essential to fostering not only good security training programs, but also creating a corporate culture in which security is valued” (<https://www.knowbe4.com/security-awareness-whitepapers>, 2018). The security that is provided out of the box, without requiring the cloud subscriber to concentrate on anything more than getting their data installed, is preferred. Default encryption in transit ensures corporate data is protected, since people are only human and prone to setting security options incorrectly when it is their responsibility. Methods to protect users from compromising themselves are also required.

4.3 Problem 3: Security Support for Cloud IaaS

In the early years at the start of the migration to the cloud, it was very difficult to get the cloud provider to address security or privacy in any corporate contracts. Today, the vendors realize that providing assurance to the customer that the cloud cybersecurity measures will be well established is a competitive advantage for acquiring a contract [13]. Companies still understand that security installation and enforcement must be the responsibility of a specialized cloud security expert, yet 64% of companies surveyed felt they needed to invest more to employ experts in the development opportunities of cloud computing. “[Fifty-two percent] of IT decision makers acknowledge that a lack of [cloud] expertise is holding

their business back” (<https://www.information-age.com/lack-cloud-expertise-loss-revenue-123468674>, accessed 3-10-2020).

5 Evaluation

The security controls for an infrastructure as a service (IaaS) environment that must be initiated by the cloud providers include data center locality, uptime/availability of the platform, encryption at rest, and encryption in transit. Other security tools that are consistent among the top three providers will also be detailed.

5.1 Data Center Locality

“The key benefits of cloud storage with a multisite infrastructure are

- an organization with multiple geographical areas,
- geolocation-sensitive data, and
- data locality and functionality close to the user”

[4]. The number of data centers each cloud provider has constructed is closely related to the longevity of the cloud offerings from those companies, with Amazon having the most data centers. The number of localities, however, is similar regardless of the longevity of the providers. Since the data is presented on world maps listing the locations and many countries have multiple data centers, these values are going to be approximate. Azure has data centers in at least 17 countries (<https://www.windowcentral.com/map-details-spread-azure-data-centers-across-world>, accessed 2-18-2020). Google is a very close second with data centers in at least 16 countries (<https://cloud.google.com/about/locations/#regions>, accessed 2-18-2020). Amazon, despite providing cloud services since 2006 [10], has the fewest with 15 different countries available for data center locations. The majority of the Amazon data centers are located in Virginia, USA (<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html#concepts-available-regions>, accessed 2-26-2020). Each provider has additional sites being built, so the number of localities is growing. All three providers support business data residency in multiple non-US locations.

5.2 Uptime/Availability of Platform

It may be startling to discover that cloud operations are not available all the time. Each provider maintains a website for reporting and mapping outages.

- Amazon – <https://outage.report/aws-amazon-web-services>
- Azure – <https://outage.report/microsoft-azure>
- Google – <https://downdetector.com/status/google-cloud/>

Per these outage reporting sites, Amazon had three outages reported on February 21, 2020. Google reported their last outage on December 7, 2019. The Azure outage report showed seven outages on February 20, 2020. These outages are reported by individuals, similar to how one calls the power company to report a power outage or to see if a known outage is affecting a particular location. Not all of the reports will be a true outage of the provider. Most just reflect the customer’s ability to interact with the cloud provider.

This confusion in outage reporting makes it difficult to compare the availability of the top three cloud environments. The reported outages could be found to be user error, home network issues, power outages at work, etc. In addition to the inability to confirm the reported outages, each cloud has data centers housed in different countries. An outage of a provider hosted in one country may or may not affect the cloud availability in another. Many companies will have the IaaS hosting environments in multiple countries.

In 2018 [14] and in 2019 [15], the year-end summary of cloud outages that affected customers gave a better explanation of actual outages for the various cloud offerings. Amazon, Azure, and Google all had major outages in both of these years. These reports also suggest that no two of the three top providers have had severe outages at the same time. Logically this would imply that the best solution for a customer that must have $24 \times 7 \times 365$ availability would be to use two or more cloud providers for the critical operations. Only one of these vendors supplied information on how distributed denial of service attacks is mitigated. “Microsoft Azure uses a novel approach of trapping [Distributed Denial of Service] DDoS attacks at the edge of its network and leaving the mission-critical machines deep inside the network available to service valid requests” [1].

5.3 *Encryption at Rest*

Data encryption for the data and backup files is necessary in case a server or hard drive is ever stolen. Unencrypted data would be readable, but encrypted data requires knowledge of the key to unencrypt. When a company sets up IaaS in the cloud, they need to be assured that the storage is protected from theft, even though there is no physical access to the server by the company.

Amazon provides Elastic Block Store (EBS) data encryption for encrypting the data and logs files of a database system. The encryption is not enabled by default and is listed as the number one security option most often misconfigured by Amazon users [12]. The options for EBS data encryption allow the customer to choose whether Amazon or the customer will provide the key for the encryption. This software allows the data to be encrypted without any required changes

to the application code. It cannot be changed once set up, however, probably contributing to the misconfiguration issues. Amazon would be smart to change this to default encryption at rest with Amazon keys and just let the customer decide on implementing a different key approach later, since it is Amazon that is at risk of the theft of the physical computers [16]. Amazon does highlight on the website that if a customer is using a nonvolatile memory express (NVMe) instance storage, then data at rest is encrypted by default [17].

Azure provides many different encryptions at rest options, including the key management option of the provider or the customer. Encryption performed by Azure is server-side encryption, but it also can use the customer's key. They also offer key encryption key (KEK) which encrypts all the encryption keys. Like Amazon, the encryption must be installed by the customer using provided graphical user interface (GUI) tools (<https://docs.microsoft.com/en-us/azure/security/fundamentals/>, accessed 2-21-2020). The keys are managed through azure active directory, which also manages the transparent data encryption (TDE) keys used to encrypt the databases.

Google is the only one of the top three providers that states in the headings of their website that encryption at rest is by default installed with the data encryption key (DEK) managed by Google (<https://cloud.google.com/security/encryption-at-rest/>, accessed 2-21-2020). The customer can switch this encryption to use a personal key instead. This decision to encrypt by default is a good decision as all data is protected even before the customer realizes that sensitive values have been added to their data model. For example, sometimes application users choose to store sensitive data such as driver's license numbers or, even worse, credit card numbers in a large character field such as "additional notes."

5.4 *Encryption in Transit*

Amazon has dozens of tools, both proprietary and third party, that will transfer or replicate data to the Amazon cloud server. Each tool would have to be evaluated separately to understand the encryption functionality provided. The Amazon Data-Sync tool does provide encryption by default for the transfer, but this is just one of the many options provided to customers for migration or replication of data to an Amazon IaaS server. This plethora of options, each with unique default settings, contributes to user uncertainty of what security options have been enacted.

Google provides a white paper that discusses the encryption in transit option for the cloud. Google defaults to encrypting all data in transit, since the networking is provided by third parties, which may not be trustworthy. The only exception is data that is in transit between servers within the physical boundaries of Google. That transit is not encrypted, but it is authenticated. The default encryption method depends on the type of connection used. An example is transfer between the customer's platform and Google Front End (GFE), which will use Transport Layer Security (TLS). There are also a number of options for use in a wide area network

(WAN) transport. Google also discusses “encryption in use” which is the encryption chosen when multiple servers are used to scale out the performance of computations (<https://cloud.google.com/security/encryption-in-transit/>, accessed 2-27-2020).

The Azure instructions for transport encryption, which must be enabled by the customer, contain extensive warnings about the effect the encryption will have on the performance of a system. The recommendation is to use a 128-bit Transport Layer Security (TLS), but a warning is issued of the performance degradation that will occur as a result of the encrypt/decrypt of packets sent. Once TLS is enabled, only servers with certificates installed on the servers can transport data, including servers used in a disaster fail-over event. In part, Azure chooses not to default to using transport encryption, as that would break the replication from an on-premise unencrypted SQL Server database to Azure. This replication is the first step in an on-premise to cloud migration of that SQL Server database with no down time to the application. Once the replication is complete, which uses log shipping from the old to the new servers, a button is clicked in Azure to finish the synchronization and complete the switch from on-premises to the cloud. So, the last step of that conversion process will need to be the addition of encryption in transport of all future data transfers (<https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/enable-encrypted-connections-to-the-database-engine?view=sql-server-ver15>, accessed 3-14-2020).

5.5 Security Common to Top Three Cloud Platforms

The top three cloud infrastructure as a service providers have many things in common. All three offer extensive security options and project the highest security standards. All provide a full-service 24 by 7 customer support center, as well as a cloud manager for each company. They all have procedures to ensure they do not allocate additional cores or memory in a way that will affect existing environments. They also all have solutions that will meet the US Health Insurance Portability and Accountability Act Business Associate Agreement (HIPAA BAA), the European Union (EU) Data Protection Directive (95/46/EC) data processing agreement, and the EU General Data Protection Regulation (GDPR) data processing agreement.

No data breach to date has been identified as being caused by a lack of security in any of these cloud providers; instead, the breaches that have occurred were a result of customer misconfiguration of the security settings. There is a concern that the employees are not acquiring the required knowledge for the best way to secure a cloud IaaS environment [10].

6 Recommendations

When a company decides to lease cloud infrastructure instead of traditional computer hardware, it comes at a cost of depending on that cloud provider to support many layers of security that were previously in the business’ control. This risk can be mitigated in several ways.

6.1 Use Multiple Cloud Providers

Just like computer hardware, cloud IaaS will have periods of unavailability, though to date these periods have not affected more than one cloud provider at any given time. The best way for a company to avoid the consequences from periods of unavailability is to distribute their compute power across more than one cloud provider. This is already happening in 84% of companies (Fig. 3) [8]. If there is an issue or contract problem with one provider, this strategy also ensures the company can be supported by a different provider. “Enterprises must always have an exit strategy that enables them to switch providers quickly. If a provider goes out of business or increases pricing to unsustainable levels, you may need to move quickly. You should always have a way to support any cloud, any time” [7].

84% of Enterprises Have a Multi-Cloud Strategy

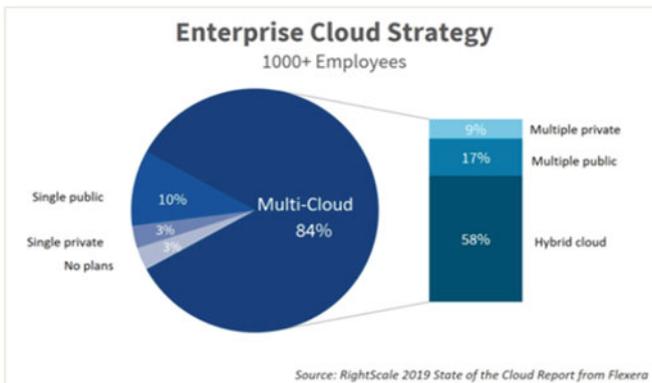


Fig. 3 84% of Enterprises Have a Multi-cloud Strategy

6.2 *Security of the Cloud: Same as On-Premises*

All the top three cloud providers have the ability to set up the identified security recommendations. Google enforces the most security out of the box, which will provide the most support to a company embracing “agile” development methods.

Whether a computer is physically in the office or hosted in the cloud, the same business security principals must be followed. These security standards must be used uniformly and enforced by a common security expert or team within the company. Prior to allowing developers access to build cloud servers, they need to be trained on how to enforce required safety procedures and standard security practices.

Never let the transfer of business data to a cloud provider be a reason to no longer monitor that environment. The IaaS servers in the cloud may reduce the necessary hardware and network resources, but it should also result in an increase in the number of cybersecurity knowledgeable personnel available to police it.

According to Mariano Mamertino, Europe, Middle East, and Africa (EMEA) economist at the global job site *Indeed*, “[f]inding, attracting and retaining tech talent is critical to business survival, and yet it is increasingly competitive for companies to find the technical talent they need as demand surges for such skill sets. Our data shows there is a global mismatch between the cloud roles advertised versus those being searched by IT professionals, which could accelerate the growth of a cloud skills gap. As this new report spotlights, there is both a financial and innovation gap to be plugged here for businesses globally” [18].

6.3 *Use Cloud Security Comparison Data*

In the “2020 Cloud Report,” created by Cockroach Labs, a series of performance tests are documented comparing the top three cloud providers Fig. 4. Reviewing the TCP-C benchmark results, “we saw that AWS [Amazon Web Services] came across on top on this benchmark once again, but that GCP [Google Cloud Provider] made tremendous strides to close the gap between itself and AWS. Azure performed similarly to the top two with its best machines. All clouds are within 5% of one another” [19, p. 33]. This graph concludes that Google is the least performant cloud provider, but in the “2020 Cloud Report” machine-type description, it is stated “[w]e expect the clouds to choose the best images for providing good performance for their VMs” [19, p. 7].

Each cloud provided different types of configuration controls and parameters, so Cockroach Labs choose to use the default settings installed by the vendor with the assumption that the cloud providers would set the defaults for optimal performance. Google, however, is the only cloud service provider to emphasize security in the default settings, by always using encryption in transit. Knowing that the TPC-C throughput test required data in transit, this security setting on the IaaS machines would throttle the performance when compared to other clouds not

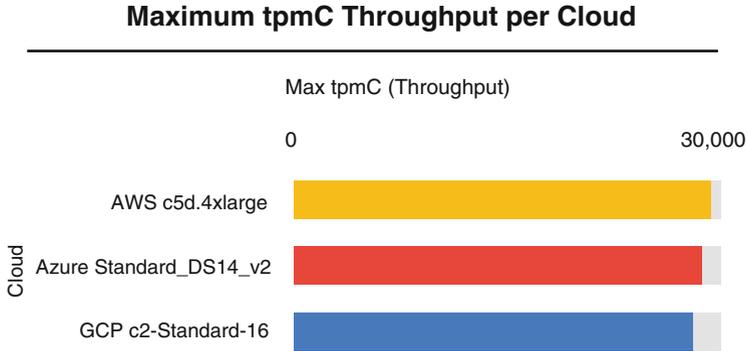


Fig. 4 2020 Cloud Report from Cockroach Labs

using encryption. Though Google was still within 5% of the performance of other cloud providers, it was the only one to secure the data in flight. It is important to be knowledgeable about each cloud’s available parameters and default settings to optimize performance of an IaaS machine while providing security for the business.

6.4 Center for Internet Security (CIS)

The Center for Internet Security (CIS) has published some excellent documents, which are free to download. These documents contain hundreds of security recommendations with detailed instructions on how to implement those recommendations for each cloud provider. This is critical information and a “must read” for any business that wants to ensure the best available security for their cloud environments.

Here is where the documents can be downloaded:

- Amazon – https://www.cisecurity.org/benchmark/amazon_web_services/
- Azure – <https://www.cisecurity.org/benchmark/azure/>
- Google – https://www.cisecurity.org/benchmark/google_cloud_computing_platform/

These are step-by-step checklists, individualized to each cloud platform, to provide the best security available on that cloud platform. The checklists are a result of years of research and experience and have consensus with the cloud providers. Every user of a cloud environment can implement the security guidelines provided in these documents with the assurance it is a proven security footprint. These are freely available and should be part of the Information Technology Security Officer’s requirements for your companies’ cloud IaaS environments.

Cloud security tools are provided for all the identified security risk areas of the top three cloud providers. Employing an Information Technology Security Officer to oversee the usage of these security tools will ensure your company runs as well

or better in cloud IaaS servers as it does on traditional onsite hardware. Proper guidelines and training will help agile developers quickly and safely use cloud environments.

References

1. S.M. Faizi, S.S. Rahman, Secured cloud for enterprise computing, in *Proceedings of 34th International Conference*, vol. 58 (2019), pp. 356–367
2. D. Gonzales, J.M. Kaplan, E. Saltzman, Z. Winkelman, D. Woods, Cloud-trust – a security assessment model for infrastructure as a service (IaaS) clouds. *IEEE Trans. Cloud Comput.* **5**(3), 523–536 (2017)
3. S.-M. Chung, M.-D. Shieh, T.-C. Chiueh, A security proxy to cloud storage backends based on an efficient wildcard searchable encryption, in *2018 IEEE 8th International Symposium on Cloud and Service Computing (SC2)* (IEEE, 2018), pp. 127–130
4. M. Kolhar, M.M. Abu-Alhaj, S.M.A. El-atty, Cloud data auditing techniques with a focus on privacy and security. *IEEE Secur. Priv.* **15**(1), 42–51 (2017)
5. P. Mell, What’s special about cloud security? *IT Prof.* **14**(4), 6–8 (2012)
6. M. Janosko, H. King, M. Saltonstall et al., Building a healthy fleet. *BeyondCorp* **43**(3) (2018). [Online]. Available: www.usenix.org
7. S.D. Lowe, *Enterprise Cloud for Dummies* (Wiley, Hoboken, New Jersey, 2016)
8. L. Dignan, Top cloud providers 2019: Aws, Microsoft Azure, Google cloud; IBM makes hybrid move; salesforce dominates SaaS, *Between the Lines* (2019)
9. M. Kan, Microsoft wins 10b pentagon cloud contract, Amazon loses out (2019). [Online]. Available: <https://www.pcmag.com/news/microsoft-wins-10b-pentagon-cloud-contract-amazon-loses-out>
10. R. Bala, B. Gill, D. Smith, D. Wright, Magic quadrant for cloud infrastructure as a service, worldwide (2019) *Gartner, July* (2019)
11. M. Younas, D.N. Jawawi, I. Ghani, T. Fries, R. Kazmi, Agile development in the cloud computing environment: a systematic review. *Inf. Softw. Technol.* **103**, 142–158 (2018)
12. C. Osborne, 99 percent of all misconfigurations in the public cloud go unreported, *Zero Day* (2019). [Online]. Available: <https://www.zdnet.com/article/99-percent-of-all-misconfiguration-in-the-public-cloud-go-unreported/>
13. K. Floyd, K. Linglebach, Perceptions of cloud storage privacy among university students. *Issues Inf. Syst.* **20**(4), 86–92 (2019)
14. H. Sarmah, Cloud outages that shook the tech world: 2018, *Analytics India Magazine* (2018)
15. S. Deoras, 8 cloud outages that shook the tech world in 2019, *Analytics India Magazine* (2019)
16. A. Degani, Amazon s3 encryption how to protect your data in s3 (2019). [Online]. Available: <https://cloud.netapp.com/blog/amazon-s3-encryption-how-to-protect-your-data-in-s3>
17. A. Namer, How to protect data at rest with Amazon ec2 instance store encryption, *Amazon EC2, How-to* (2017). [Online]. Available: <https://aws.amazon.com/blogs/security/how-to-protect-data-at-rest-with-amazon-ec2-instance-store-encryption/>
18. N. Ismail, Is a lack of cloud expertise causing a loss of revenue? *Information Age* (2017). [Online]. Available: <https://www.information-age.com/lack-cloud-expertise-loss-revenue-123468674/>
19. P. Bardea, C. Dillon, N. VanBenschoten, A. Woods, 2020 cloud report (2020). [Online]. Available: <https://www.cockroachlabs.com/guides/2020-cloud-report/>

Orientation and Line Thickness Determination in Binary Images



Sean Matz

1 Introduction

Inverse problems involve determining an unknown quantity (a cause) associated with a particular object based upon measurements of the effects associated with this object. An inverse problem starts with observations of effects and determines the causes. Therefore, the concepts in inverse problems give rise to an underlying theory for remote sensing and nondestructive testing. Inverse problems in the areas of tomography, pattern recognition, computerized geophysical tomography (CGT), computed axial tomography (CAT), barcode scanners, electron microscopy, and seismology involve the Radon transform. The solution to many inverse problems utilizes the Radon transform. The Radon transform in two dimensions is the integral transform which takes a function f defined on the plane to a function Rf defined on the two-dimensional space of lines in the plane whose value at a particular line is equal to the line integral of the function over that line. The Radon transform represents a transformation from a function of rectangular coordinates x and y to a function of coordinates ρ and θ . In 2D, the Radon transform can be expressed using a two-dimensional delta function as

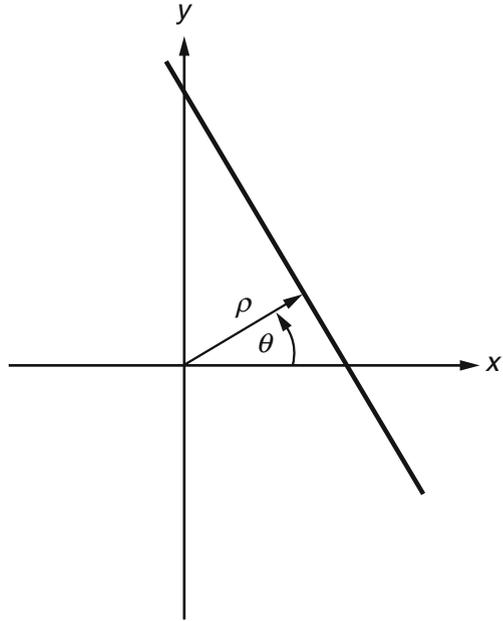
$$R(\rho, \theta) = \int_{R^2} f(x, y) \delta(\rho - x \cos \theta - y \sin \theta) dx dy$$

where

$$\rho = x \cos \theta + y \sin \theta$$

S. Matz (✉)
Claremont Graduate University, Claremont, CA, USA
e-mail: smatz@att.net

Fig. 1 Geometry of summation line in image



is the line along which f is integrated to produce the line integral which is $R(\rho, \theta)$. The same expression above used to define the Radon transform $R(\rho, \theta)$ is also used to define the projection $P_\theta(\rho)$.

If $f(x, y)$ is a 2D image intensity function, computation of its Radon transform yields the projections across the image at varying orientations and distances (from the origin) ρ . The Radon transform maps lines in image space to points in feature space. Bright (dark) lines in an image are mapped by the Radon transform to bright (dark) points in feature space as mentioned by Murphy in [1]. For each angle θ and each distance ρ , the intensity of the object through which a ray perpendicular to the ρ axis passes is added up at $P_\theta(\rho)$. Figure 1 shows the typical geometry of a line along which the intensity is summed to produce a Radon transform. A set of many such projections under different angles forms a sinogram.

2 Previous Work

The paper [2] by Jafari-Khouzani and Soltanian-Zadeh presents a method for addressing the problem of rotation invariant texture classification. For directional textures, the wavelet features must be computed for a particular direction. In this paper, the Radon transform is first utilized to determine the primary texture direction. The texture is then rotated in such a way that its primary direction is at 0 degrees.

The paper [3] by Aggarwal and Karl notes that the determination of the location and orientation of straight lines in images is of primary interest in fields like computer vision and image processing. The Hough transform (a special case of the Radon transform) has been employed to address this problem for binary images. The authors of this paper consider the line detection problem in images as an inverse problem. They make use of the inverse Radon operator which relates parameters involving the line location and orientation to the noise-degraded image. This places the problem within a regularization context and improves the performance of Hough-based line detection through the use of prior information with respect to regularization.

The paper [4] by Rajput, Som, and Kar uses the Radon transform to determine the orientation of license plates. There, each image is of a license plate oriented at a particular angle. They don't, however, deal with multiple license plates at different orientations in a single image. They briefly discuss other methods which attempt to address multiple orientations and their limitations.

The authors of [5] utilize the Radon transform to determine the orientation of fingerprints as part of a fingerprint recognition system. They note that the orientation of the ridge and the valley in a fingerprint is very important in the identification of fingerprints. The Radon transform is employed for obtaining this orientation information. They computed projections at seven different angles using the Radon transform for a given fingerprint image and did not consider an image with multiple fingerprints. They never tried to analyze multiple fingerprints in a single image.

In [6], the authors analyze thick line center and width estimation. They present a method to properly detect thick lines and their centers using mathematical morphological operators.

In [7], the authors consider a new straight line detector in gray scale images where line segments are set up with a thickness parameter intended to provide a quality measure of the detected feature.

In [8], the authors show that a convolutional network can learn subtle features to estimate the orientation of images.

In [9], the author proposes a technique for determining the orientation angle of text using the Radon transform.

In [10], the authors discuss a method to obtain orientation of text in a document image as well as to place the image at the proper orientation. This is based on pattern recognition where a particular pattern has been discovered by extensive studies of different scripts.

In [11], the researchers propose a technique for determining the camera orientation in a vehicle by obtaining three orthogonal vanishing points. The vanishing point along the driving direction is first estimated. Then this vanishing point is obtained by using the Hough transform. The remaining vanishing points are selected from the circular histogram, and they are orthogonal to the vanishing point along the driving direction. Finally, the orientation is estimated using the three orthogonal vanishing points.

In [12], the authors propose a line detection technique using the first derivative of the Gaussian probability density function (pdf). This method can, at the same time, estimate line width.

In [13], the authors discuss a technique for detecting vehicles using a template-based directional chamfer matching approach and vehicle orientation estimation using a refined segmentation and then a new Radon transform-based variance peak detection.

In [14], the researchers describe a method for detecting texture orientation. Since the DCT and FFT have a high computational cost, the authors applied the DWT and the Radon transform, taking little CPU time.

In [15], the chapter describes the development of methods for orientation analysis. Enhanced orientation analyses with other techniques in image analysis such as porosity and multispectral methods are described.

In [16], the authors utilize a Canny edge detector to find the image edges. Then, they combine a Hough transform with the method of least squares to solve the problems of the standard Hough transform. Finally, they present a new model based on the imaging principle of a camera for icing thickness determined by employing the radius of the transmission line as a reference. However, the authors don't consider more than one transmission line in an image.

In [17], the authors note that the standard Hough transform does not yield the length and width of a line segment detected in an image. Their method utilizes a statistical analysis of voting cells around a local maximum in Hough space. In image space, voting cells and voting values are analyzed. A relationship between the voting variance and the voting angle is determined. This relationship is expressed by a quadratic polynomial. In Hough space, statistical variances of columns around a peak are calculated and then fit to a quadratic polynomial. The length and width of a line segment are determined by solving equations generated by comparing the corresponding coefficients of two functions. The algorithm was a great modification of the Hough transform to determine line thicknesses. However, the authors tested the algorithm on highway lane stripes that were thick and far apart as opposed to thin lines spaced closely together.

In [18], the authors investigate many problems that affect centerline detection using the Radon transform. A mean filter is employed to find the peak in a Radon transformed image, and a profile analysis method is utilized to obtain a better estimate of the line parameters. Experiments show that this method is effective in finding the centerline and estimating the line width of thick lines.

3 Orientation Determination

The orientation detection algorithm computes the Radon transform (sinogram) of an image of lines at various angles. Then, the maximum value of the sinogram is computed at each angle and is plotted as a function of angle. This allows for the selection of a threshold that will separate the actual Radon transform-detected angles from the rest of the values in the sinogram or the clutter level. The clutter in the image is caused by the number of lines at different angles being in close proximity to each other as they emerge from one point. The clutter level is evident when the histogram of the maximum values of the sinogram is plotted. The fact that

the maximum sinogram value obtained at the angle of each line in the image is much greater than the level of the clutter allows for the selection of a threshold that will separate the actual Radon transform-detected angles from the clutter values in the sinogram. Finally, the mean and the standard deviation are computed. The column index in the sinogram denotes the angle in degrees.

4 Determination of Line Thickness

The line thickness algorithm determines the thickness of lines at selected angles by considering the pattern of the pixels of those lines. The line thickness algorithm consists of four parts, a part to detect horizontal lines and determine their thickness, a part to detect vertical lines and determine their thickness, a part to detect lines at 45 degrees and -45 degrees and determine their thickness, and parts to determine lines at other selected angles.

To detect a horizontal line, the algorithm determined that a pixel located at (i, j) was on a horizontal line if the image intensity $I(i, j) \neq 0$ and if $I(i, j - 1) = 0$ and if $I(i, j + 1) \neq 0$ and if $I(i - 1, j) = 0$ and if $I(i, j + 5) \neq 0$ and if $I(i, j + 10) \neq 0$, where I denotes the pixel value or intensity. Then, to determine the thickness, while $I(i + k, j) \neq 0$, increment k ($k = k + 1$) where k is the row increment. The final value of k denotes the thickness.

A pixel at location (i, j) was determined to lie on a vertical line if the image intensity $I(i, j) \neq 0$ and if $I(i - 1, j) = 0$ and if $I(i, j - 1) = 0$ and if $I(i + 1, j) \neq 0$ and if $I(i + 5, j) \neq 0$ and if $I(i + 10, j) \neq 0$. Then, to determine the line thickness, while $I(i, j + k) \neq 0$, increment k ($k = k + 1$). The final value of k denotes the thickness.

A pixel at location (i, j) is determined to lie on a line (which is increasing with j) at an angle of multiple pixel thickness if the image intensity $I(i, j) \neq 0$ and if $I(i, j - 1) = 0$ and if $I(i, j + 1) = 0$ and if $I(i + 1, j) \neq 0$ and if $I(i + 1, j + 1) \neq 0$ and if $I(i + 1, j + 2) = 0$. To determine the line thickness, while $I(i, j + k) \neq 0$, then increment k ($k = k + 1$). The final value of k denotes the thickness.

A pixel at location (i, j) is determined to lie on a line (which is decreasing with j) at an angle of multiple pixel thickness if the image intensity $I(i, j) \neq 0$ and if $I(i, j - 1) = 0$ and if $I(i, j + 1) = 0$ and if $I(i + 1, j - 1) \neq 0$ and if $I(i + 1, j) \neq 0$ and if $I(i + 1, j + 1) = 0$. To determine the line thickness, while $I(i + k, j) \neq 0$, then increment k ($k = k + 1$). The final value of k denotes the thickness.

5 Results for Orientation Determination

This algorithm was tested on an image containing 14 different angles and on an image containing lines at 35 different angles.

Figure 2 shows the input image with 14 lines at different angles, starting at 5 degrees and continuing in 5-degree increments up to 70 degrees. Figure 3 shows a

histogram of the maximum of the sinogram at each angle. What appear clearly are 14 different angles associated with lines at the original 14 angles in the input image.

Figure 4 shows the input image with 35 lines at different angles (starting at 2 degrees, with 2-degree increments, up to 70 degrees). Figure 5 shows a histogram of the maximum of the sinogram at each angle. Thirty one of 35 angles can be

Fig. 2 Input Image with Lines at 14 Different Angles

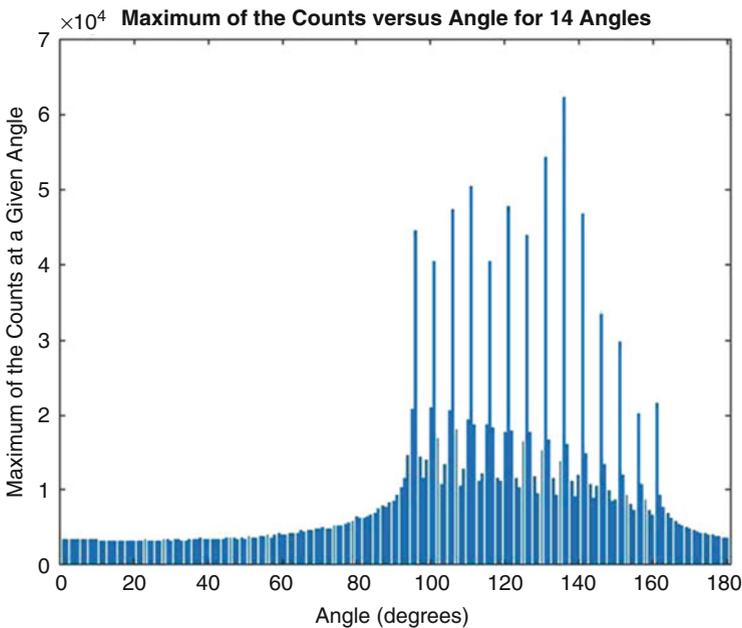
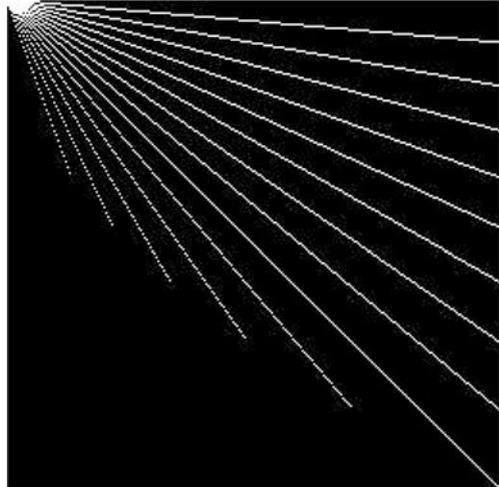


Fig. 3 Histogram of Maximum Sinogram at each Angle for 14 Input Angles

counted from the histogram. However, the sinogram of the image shown in Fig. 6 shows 35 peaks which correspond to 35 detected lines at 35 different angles.

The mean of the 14 detected angles was computed to be 37.5 degrees which is exactly the mean of the 14 input angles. Similarly, the mean of the 35 detected

Fig. 4 Input image showing lines at 35 different angles

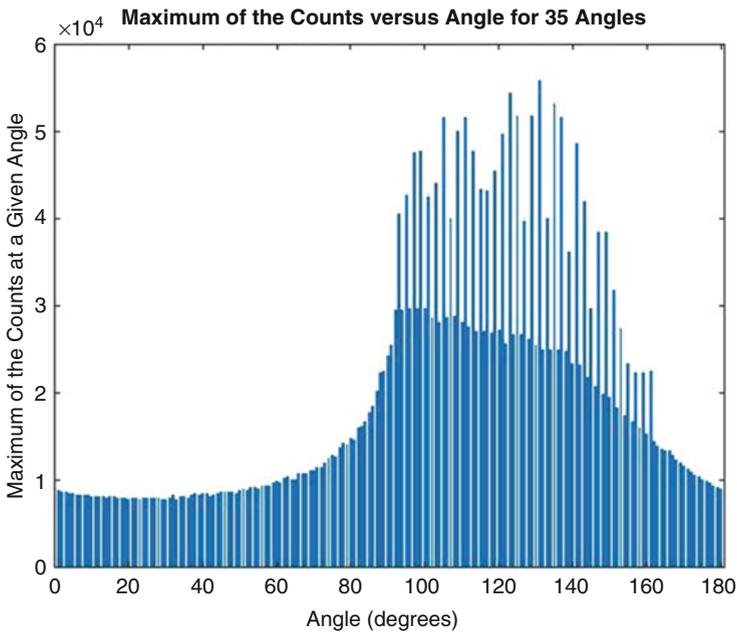
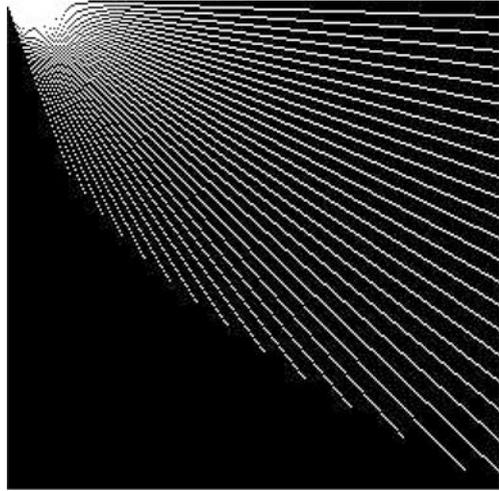


Fig. 5 Histogram of maximum of sinogram at each angle for 35 input angles

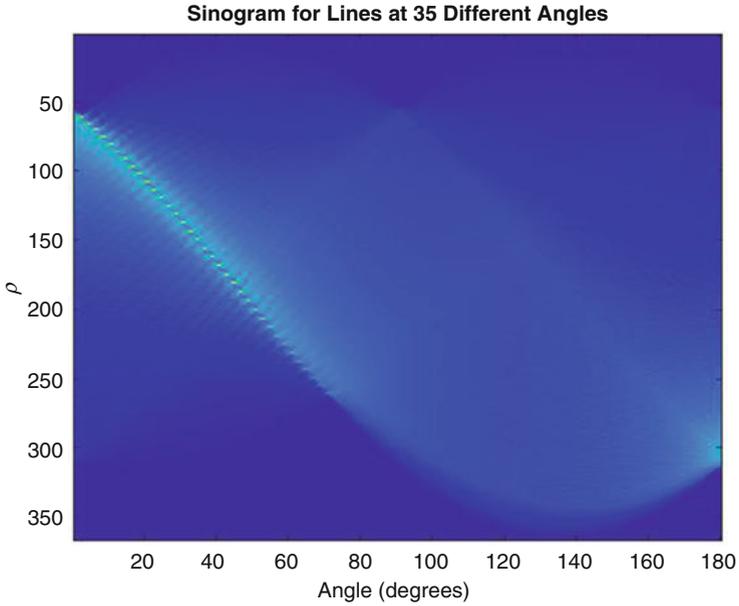
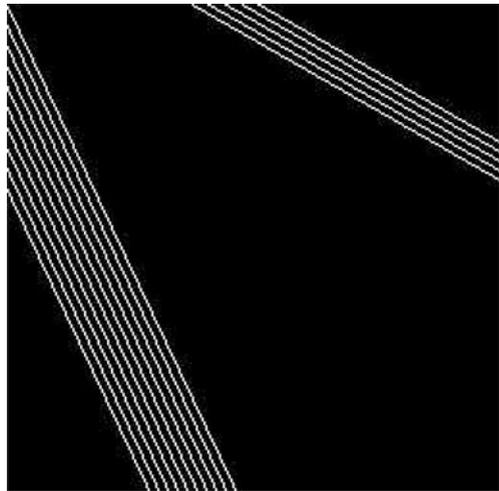


Fig. 6 Sinogram of 35 lines at 35 angles showing the peaks of the 35 detected lines

Fig. 7 Image with 10 lines at 25 degrees and 5 lines at 60 degrees, with angles measured with respect to the vertical



angles was computed to be 36 degrees which is precisely the mean of the 35 input angles.

Figure 7 shows an image with 10 lines at an angle of 25 degrees and 5 lines at an angle of 60 degrees, where the angles are measured with respect to the vertical edge of the image.

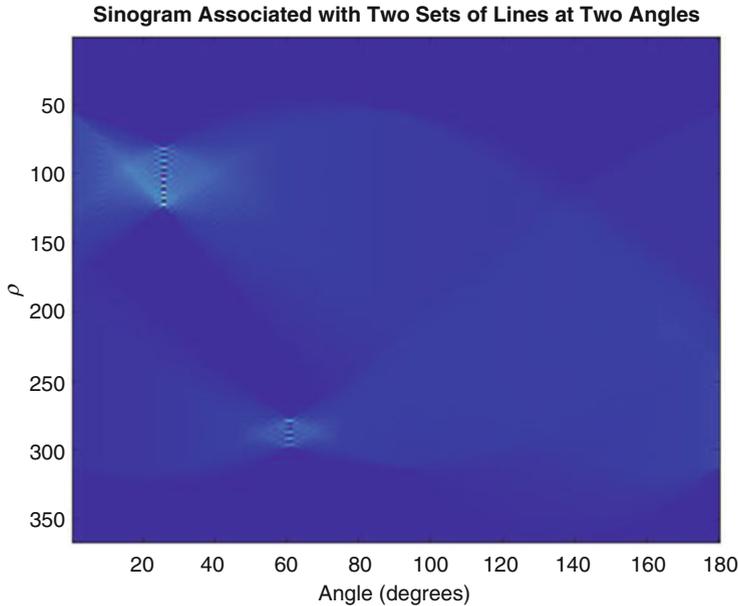


Fig. 8 Sinogram showing peaks at two different angles

When the Radon transform processed the image in Fig. 7, it should produce peaks at the two angles, equal to the number of lines at each angle. The resulting sinogram shown in Fig. 8 shows the peaks at the angles 25 degrees and 60 degrees, exactly what the theory predicts.

6 Results for Line Thickness Determination

The input image in Fig. 9 shows eight lines of varying thicknesses. The far-right vertical line has a thickness of 5 pixels. The vertical line in the middle of the image has a thickness of 2 pixels. The far-left vertical line has a thickness of 3 pixels. The top horizontal line has a thickness of 4 pixels. The horizontal line in the middle of the image has a thickness of 1 pixel. The bottom horizontal line has a thickness of 2 pixels. The diagonal line that is decreasing from left to right has a thickness of 3 pixels. The diagonal line that is increasing from left to right has a thickness of 4 pixels. The sum of all of the pixel widths is equal to 24. The number of lines is 8. Therefore, the average pixel thickness is 3 with a standard deviation of 1.2247. This is exactly what the MATLAB code determined.

The input image in Fig. 10 shows 24 lines of different thicknesses used as input into the line width algorithm. In addition to the horizontal and vertical lines, there are lines at angles of 5, 35, 40, 45, 50, and 55 degrees. Each line decreasing from

Fig. 9 Input Image with eight lines of different thicknesses

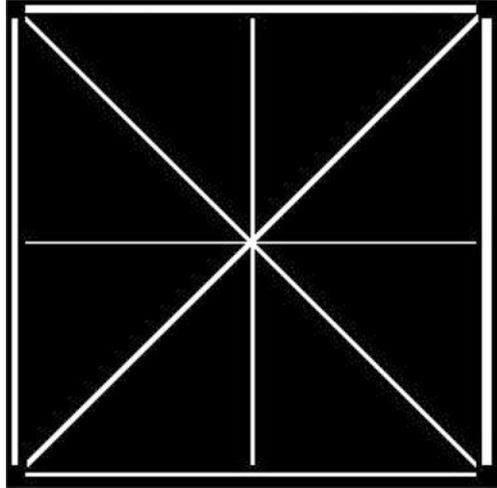
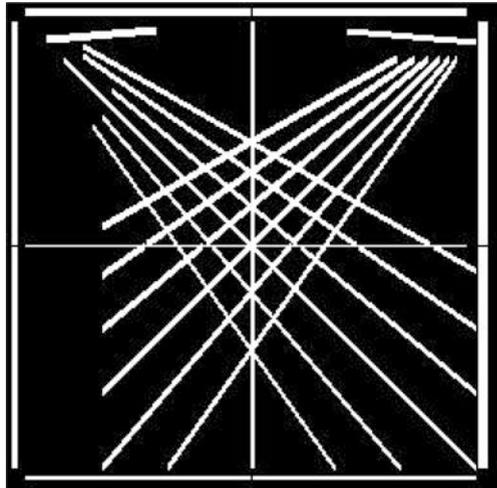


Fig. 10 Input image with 24 lines of different thicknesses



left to right had a width of 3 pixels, and each line increasing from left to right had a width of 4 pixels. The algorithm determined that the average thickness is 3.333 pixels and that the standard deviation is 0.942809 pixels.

7 Conclusions

In this work, I considered two related problems, orientation detection of lines and determination of line thickness. The use of the Radon transform allowed for the recovery of the angles of lines in an input image. The peaks denoting the points in Radon space corresponding to the lines in image space could be clearly seen at

the angles which were the same as the input angles of the lines. While the peaks corresponding to the lines at different angles may not always appear in a bar plot of the maximum sinogram values (as in the case of 35 lines at 35 different angles), the sinogram will usually show those distinct peaks. The only exception to the sinogram not displaying those peaks is when a line does not extend across a sufficient part of the image. The use of different test images, which explored different aspects of the orientation algorithm, illustrated properties of the Radon transform like the mapping of a line to a point, and that parallel lines will map to points at the same angle in Radon space but be displaced by the amount of separation between the lines. The sinogram which is the Radon transform of an input image clearly displayed peaks in Radon space which were points to which lines in image space at different angles were mapped by the Radon transform. The Radon transform detects lines whose length is a significant part of one of the image dimensions. Thus, as long as a line extends along most of one of the image dimensions, the orientation algorithm will detect the angle of that line.

The line width problem considered parallel lines at given angles as well as many lines at different angles. In the line width problem, the algorithm was able to determine the thicknesses of lines at selected angles. In addition to testing on 8, 12, 18, and 24 lines at different angles (with the results of 8 and 24 lines shown above), this algorithm was also tested on an image with seven parallel lines with four different thicknesses, and it correctly detected the different thicknesses of lines. Although the line width algorithm was tested on images with a maximum line width of 5 pixels, the implementation of this algorithm allows for lines of any width.

The practical applications of these algorithms are the following: road network extraction from high-resolution remotely sensed imagery, detection of text of arbitrary orientation, and ability to determine crack thickness and orientation.

Another application of orientation detection is that of detecting text of an arbitrary orientation in an image obtained from a mobile computing device.

An additional application of the orientation and line with detection algorithms is that of line detection and tracking for an advanced driver assistance system.

References

1. L.M. Murphy, Linear feature detection and enhancement in noisy images via the Radon transform. *Pattern Recogn. Lett.* **10**, 279–284 (1986)
2. K. Jafari-Khouzani, H. Soltanian-Zadeh, Radon transform orientation estimation for rotation invariant texture analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(6), 1004–1008 (2005). <https://doi.org/10.1109/TPAMI.2005.126>
3. N. Aggarwal, W.C. Karl, Line detection in images through regularized Hough transform. *IEEE Trans. Image Process.* **15**(3), 582–591 (2006). <https://doi.org/10.1109/TIP.2005.863021>
4. H. Rajput, T. Som, S. Kar, Using radon transform to recognize skewed images of vehicular license plates. *Computer* **49**(1), 59–65 (2016). <https://doi.org/10.1109/MC.2016.14>
5. D. Tiwary, K. Srinivasulu, Anveshraj, Fingerprint identification using radon transform. *Int. Refereed J. Eng. Sci. (IRJES)* **3**(10), 106–110 (2014)
6. M. Aleman-Flores, L. Alvarez, P. Henriquez, L. Mazon, Morphological thick line center detection, in *Image Analysis and Recognition. ICIAR 2010*, ed. by A. Campilho, M. Kamel.

- Lecture Notes in Computer Science, vol. 6111 (Springer, Berlin/Heidelberg, 2010). <https://doi.org/10.1007/978-3-642-13772-3>
7. P. Even, P. Ngo, B. Kerautret, Thick line segment detection with fast directional tracking, in *Image Analysis and Processing – ICIAP* (2019). <https://hal.archives-ouvertes.fr/hal-02189916>
 8. P. Fischer, A. Dosovitskiy, T. Brox, Image orientation estimation with convolutional networks, in *Pattern Recognition. DAGM 2015*, ed. by J. Gall, P. Gehler, B. Leibe. Lecture Notes in Computer Science, vol. 9358 (Springer, Cham, 2015). <https://doi.org/10.1007/978-3-319-24947-6>
 9. A.M. Khidhir, Use of radon transform in orientation estimation of printed text, in *The 5th International Conference on Information Technology (ICIT)* (2011)
 10. L. Kumari, S. Debbarma, R. Shyam, Text orientation detection from document image of Indian scripts. *Singaporean Journal Scientific Research (SJSR)*. **3**(1), 146–149 (2010)
 11. Y. Jo, J. Jang, M. Shin, J. Paik, Camera orientation estimation using voting approach on the Gaussian sphere for in-vehicle camera. *Opt. Express* **27**(19), 26600–26614 (2019)
 12. Q. Li, L. Zhang, J. You, D. Zhang, P. Bhattacharya, Dark line detection with line width extraction, in *International Conference on Image Processing (ICIP)* (2008)
 13. R. Pelapur, F. Bunyak, K. Palaniappan, G. Seetharaman, Vehicle detection and orientation estimation using the radon transform. *Proc. SPIE* **8747** (2013). <https://doi.org/10.1117/12.2016407>
 14. S. Sahu, S.K. Nanda, T. Mohapatra, Digital image texture classification and detection using radon transform. *Int. J. Image Graph. Sig. Process. (IJIGSP)* **5**(12), 38–48 (2013)
 15. N. Keith Tovey, M.W. Hounsflow, J. Wang Orientation analysis and its applications in image analysis. *Electron Phys.* **93**, 219–329 (1995)
 16. J. Wang, J. Wang, J. Shao, J. Li, Image recognition of icing thickness on power transmission lines based on a least squares Hough transform. *Energies* **10**, 415 (2017). <https://doi.org/10.3390/en10040415>
 17. Z. Xu, B.-S. Shin, R. Klette, Determination of length and width of a line segment by using a Hough transform, in *DCCI 2014*. LNCS 8668 (Springer International Publishing, Cham, 2014), pp. 190–201
 18. Q. Zhang, I. Couloigner, Accurate centerline detection and line width estimation of thickness using the radon transform. *IEEE Trans. Image Process.* **16**(2), 310–316 (2007)

Greedy Navigational Cores in the Human Brain



Zalán Heszberger, András Majdán, András Biró, András Gulyás,
László Balázs, Vilmos Németh, and József Biró

1 Introduction

Greedy navigation is a type of hop-by-hop routing strategy in geometrically embedded networks. Geometric embedding means that nodes have either physical coordinates (e.g., 2D/3D Euclidean or spherical) or inferred coordinates in a more abstract metric space (e.g., hyperbolic space). The operation is as follows: along the greedy route, every node passes the information to that neighboring node which is closest (and closer than the passing node) to the destination node in the metric space in which the network is embedded. Clearly, nodes have to know only the coordinates of their neighbors; based on these and the destination node coordinate (available typically with the information packet arrived), the distances can be calculated, and the next hop as being closest to the destination can be identified. This is a simple operation using only local information providing distributed operation, that is why this type of routing is an actively researched area both in technological, natural, and social networks.

Z. Heszberger · A. Majdán · A. Gulyás

Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest, Hungary

MTA-BME Information Systems Research Group, Budapest, Hungary

e-mail: heszi@tmit.bme.hu

A. Biró · L. Balázs · J. Biró (✉)

Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest, Hungary

e-mail: biro@tmit.bme.hu

V. Németh

BME Center of University-Industry Cooperation, Budapest, Hungary

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_25

Real networks having coordinates in a metric space are not necessarily 100% greedy navigable. It means that greedy routing can get stuck in a node which has no direct link to the destination node and all of their neighbors are farther from the destination. Nevertheless, the level of greedy navigability (even if it is far from 100%) may be an important sign of the coevolution of the function structure of the network and the efficiency/inefficiency of the metric space. For example, in [1], Gulyás et al. have tested several real networks against greedy navigability and found that scale-free networks (Internet, US airport networks, metabolic networks, word networks) are highly navigable (80–90%) with 2D hyperbolic space and the non-scale free Hungarian road networks is well greedy navigable in the 2D Euclidean space (84%). For this, our fundamentally new, so-called function-structure approach was to create first the minimalistic and 100% greedy navigable skeleton, the so-called Greedy Navigational Core (GNC) network, and then to test the presence of these cores in real networks. The GNC is a result of solving heuristically the hard optimization problem of finding the minimum set of links providing maximum (100%) greedy navigability. The GNC can also be obtained by the so-called network greedy navigation game. In this game, nodes have the strategies to set up minimum number of links by which they can cover all non-neighboring nodes by greedy next-hop forwarding. The Nash equilibrium of this game (previously referred to as Nash Equilibrium Network, NNG network) is the Greedy Navigational Core [1]. In the last couple of years, thanks to the advancement of MRI-based imaging technologies, brain structural networks have been widely investigated [2–4]. For setting up these networks, the brain cortex has been divided into parcels (ROI (region of interest)) acting as nodes, and (as a possible method) MRI-based diffusion spectrum imaging (DSI) is used to explore the nerve fiber paths (travelling through the white matter) connecting the ROIs. The brain parcels have inherently 3D Euclidean coordinates which provide the possibility of testing the Euclidean greedy navigability of such networks. We have shown first in [1] that the Greedy Navigational Core as minimal network that is maximally navigable by design presents substantially also in a five subject-based averaged structural brain network. More specifically, it is found that the GNC precision (the ratio of the number of the GNC links included in the real network and the total number of GNC links, sometimes referred to as true positive rate) is 89% in this brain network. The GNC for this brain network was created by using only the 3D Euclidean coordinates of brain parcels; no other information was used. Subsequent studies have advocated thoughts and results on greedy navigation of brain networks [5–9]. They highlighted that greedy navigation as a decentralized communication strategy is well suited to spatially embedded networks like brains. In [6, 7], the authors have followed the well-established structure-function approach, namely, structural brain networks have been directly tested against the function greedy navigation in terms of different success measures [10]. The authors of [8] have explicitly used our unorthodox function->structure method [1] to generate Greedy Navigational Cores (referred to as Nash Equilibrium Network, NNG network) and used them to predict resting-state functional connectivity with high accuracy. In this study, using the function-structure approach, we present detailed

and elaborated results on structural greedy navigability of networks of human brain in five different scales, including the consistency, robustness, and structural similarities.

2 Results

2.1 Individual Networks

We have performed investigations on structural greedy navigability as the level of GNC precisions in 200 structural brain networks from 40 individual subjects at 5 different scales (these scales correspond to resolutions of 83, 129, 233, 463, 1015 nodes in the brain structural networks) [11]. More details on the dataset are in subsection Methods. For the GNC network generations, only the physical (3D Euclidean) coordinates of the brain parcels were used; no other anatomical data or considerations were utilized. First we present results on the original networks inferred, without any link removal (pruning). We found that the level of GNC precision (the ratio of GNC links being also in the brain network) is high and quite consistent (relatively low standard deviation around the mean) among the 40 brain networks within all scale, in spite of the fact that the 40 brain networks significantly differ from each other at all scales. The mean (and the standard deviation) of the GNC precisions (besides some other network parameters) in different scales (with increasing resolutions) are in Table 1.

Pruning by anatomical strength Now we turn to the case of network pruning, that is, from the original networks, links with low weights (possibly spurious nerve fiber paths) have been sequentially removed and these pruned networks have been tested against greedy navigability. For this, in every scale, a sequence of networks has been generated using different weight thresholds for pruning. Table 2 summarizes the level of inclusion of GNC in the pruned networks for some characteristic threshold values. For instance, when more than 50% of the links are removed from the brain networks, the precisions remain close to the original values. One can also

Table 1 Mean and standard deviation of GNC precisions on different scales of brain structural networks

	Scale1	Scale2	Scale3	Scale4	Scale5
# Nodes	83	129	233	463	1015
Average # links in brain networks	1119.4	1975.5	3799.27	7246.48	14254.8
Average # links in GNC	177.7	292.9	553.4	1153.3	2656.6
Average degree in brain networks	26.97	30.63	32.61	31.30	28.09
Average degree in GNC	4.28	4.54	4.74	4.98	5.24
Mean of GNC precision	0.85	0.88	0.81	0.70	0.51
Standard deviation of GNC precision	0.025	0.019	0.017	0.017	0.025

Table 2 GNC precisions (mean and standard deviation) in pruned networks. The first row shows the case without pruning

Thresholds	% of links removed	Scale1	Scale2	Scale3	Scale4	Scale5
0	0%	0.85(0.025)	0.88(0.019)	0.81(0.017)	0.70(0.017)	0.51(0.025)
10^{-5}	5–10%	0.84(0.024)	0.88(0.019)	0.81(0.017)	0.69(0.017)	0.51(0.025)
10^{-4}	23–29%	0.81(0.025)	0.85(0.018)	0.79(0.016)	0.68(0.016)	0.50(0.024)
10^{-3}	50–55%	0.72(0.031)	0.75(0.019)	0.68(0.020)	0.57(0.019)	0.42(0.021)
10^{-2}	82–84%	0.39(0.034)	0.37(0.033)	0.34(0.028)	0.27(0.022)	0.18(0.017)

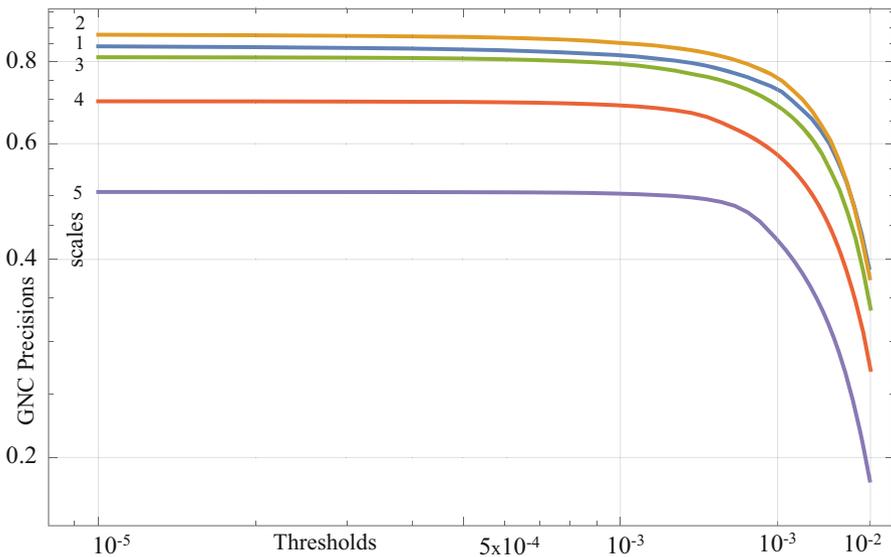


Fig. 1 GNC Precision in the function of weight thresholds of pruning

observe that larger part of the navigational core is missing from higher-resolution brain networks; however, the precisions are still consistent according to the low standard deviations. A more detailed view of the effect of the network pruning can be seen in Fig. 1. One can observe that up to a certain threshold, the GNC precision remains almost intact. Then further increasing the number of links cut out, the GNC precision starts to decrease; however, even in case of extremely pruned networks, it remains exceptionally high. This observations support the hypothesis that the GNC mostly consists of anatomically strong links and spurious and/or anatomically weak connections are less likely part of the navigational cores.

Pruning by link prevalence score Besides anatomical weighting strategies, the link prevalence score (the number of networks containing the link) can also be used to identify possibly existent (high prevalence score) and possibly nonexistent (low prevalence score) links in the inferred brain structural networks, and based

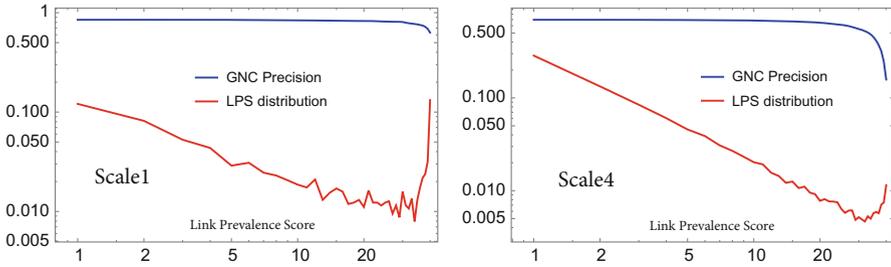


Fig. 2 Greedy Navigational Core precision and link prevalence score distribution in the function of link prevalence score

on this, one can compromise the false positives and false negatives in pruning the networks [12]. Here, GNC precisions are also presented in *individual* networks (the abovementioned 40-subject, 5-scale networks with the number of nodes 83, 129, 234, 463, 1015 in scales 1, 2, 3, 4, 5, respectively) thresholded by the link prevalence scores. Thresholding means in this case that in an individual network, only that links are kept, which are present at least $T - 1$ in other networks too, where T is the threshold. Within a resolution, every network is thresholded by all possible values of LPSs (1, . . . , 40), and then the GNC precisions are measured in all resulted networks (this corresponds sequence of 1600 networks in every scale). The GNC precisions are then averaged over the subjects for every LPS threshold. The important observation is that GNC precisions are still consistent (low variations across subjects) and robust against LPS thresholding in all scale. For example, in Scale 1 for LPS = 1 (no link is removed), GNC precision is 0.85 and for LPS = 30 (about 32% of links are removed from every network), GNC precision is still as high as 0.80. In all the five scales, it can be observed that for lower values of LPS thresholds, the GNC precision remains almost intact, while for higher values, its decrease is fastening. The fastening decrease measurably coincides with the right-hand side (consisting of possibly existing links) of the link prevalence distribution (see Fig. 2). This means that most of the true positive links in GNC networks are also possibly existent in the brain networks.

2.2 Average Networks

In Scale 5, we have constructed a sequence of average networks, based on averaging networks over link weights (inferred from measured anatomical strengths of fiber paths), and cutting out links with small average weights. We have also generated an average GNC network based on averaging the centroids of the brain parcels. The size of the average GNC network (the number of links) is 2652. Regarding

the average network sequence, the following table shows the GNC precision in the function of the size of the average network:

Threshold	10^{-8}	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}
Averaged Brain network size	101,468	101,450	95,929	67,553	39,406	15,996	2381
GNC Precision	0.925	0.925	0.924	0.907	0.874	0.769	0.263

Note that if none of the links have been cut out, the average brain network contains 101,468 links (this corresponds to a 19.7% connection density, because full mesh would have 514,605 links) and in this case, 0.9253 fraction of the 2652 links of the average GNC is in this network. If the number of links in the average network is quite comparable to the sizes of individual networks (threshold=0.001; size of average network = 15,996; 3% connection density), the GNC precision is still as high as 76.9%. When the size of the average network is so extremely small that is comparable to the average GNC network, the GNC precision is still amazingly 26.28%.

2.3 Greedy Frames

The Greedy Navigational Core as the Nash equilibrium of the greedy network formation game is not unique. We always choose that one among these minimalistic networks which has the lowest aggregate link lengths. Nevertheless, there is always a common subset of the Nash equilibria called Greedy Frame [1]. These common links appearing in all maximally navigable minimal networks can be identified as follows: Let us take two nodes in the network, u and v . If v is the closest node to u then for ensuring 100% greedy navigability, the (directed) link $v \rightarrow u$ must exist. Otherwise, there would be no greedy path from v (or through v) to u . The Greedy Frames can also be determined in our brain networks purely by using only the 3D coordinates of the parcels. The mean values (and standard deviation) of the Greedy Frame precisions (without pruning) in Scale 1 to Scale 5 are 0.942(0.022), 0.985(0.024), 0.947(0.026), 0.832(0.031), and 0.595(0.034), respectively. One can observe that these inclusion ratios are even higher than the GNC precisions keeping the low variability within the scales. The inclusion ratio of the Greedy Frames is also robust against pruning. Figure 3 shows that when cutting out links up to LPS = 10, the Greedy Frame precisions do not change significantly in all scales. In case of lower resolutions (the first three scales), it is even true for LPS = 30. In the highest resolution (Scale 5), the Greedy Frame inclusion is remarkably lower than in other scales, but its decrease is more flat between LPS = 20 and LPS = 40.

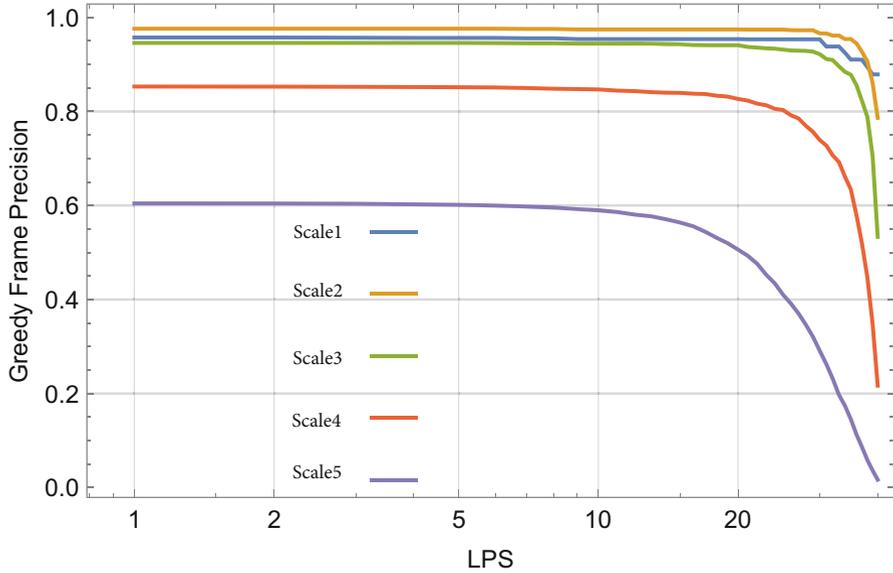


Fig. 3 Greedy Frame precision in the function of link pruning values based on link prevalence score

3 Discussion

The consistency and robustness of the Greedy Navigational Core network precisions are remarkable in the light that these artificially generated networks are inferred only from the physical coordinates of the brain parcels by pure geometric computations and optimizations; no other anatomical data or consideration were used. The 40 brain networks within a scale show significant differences in terms of the links between the brain parcels. These differences originate from two main sources, measurement/imaging inaccuracy and anatomical variability. The effects of these can be decreased by pruning the networks (decreasing the false-positive rates due to inaccurate imaging) and determining the optimal brain parcellation and parcel coordinates according to individual anatomy of subjects. As the results showed, the GNC precision is significantly robust against pruning. This means that the common links of GNC and brain networks are likely existent and not spurious due to imaging artifacts. The Greedy Navigational Cores can also trace the variations of brain networks through changes of the brain parcel coordinates, which results in keeping GNC precisions at high level with low variations.

GNC networks are not generative models for predicting brain links, because this is a minimalistic network containing much less links than the underlying brain network. This is also reflected by the average degrees, that is, the GNC average degrees are between 4.2 and 5.3, while the brain network degrees lie between 26 and 32. In spite of this fact, the GNC networks show structural similarities

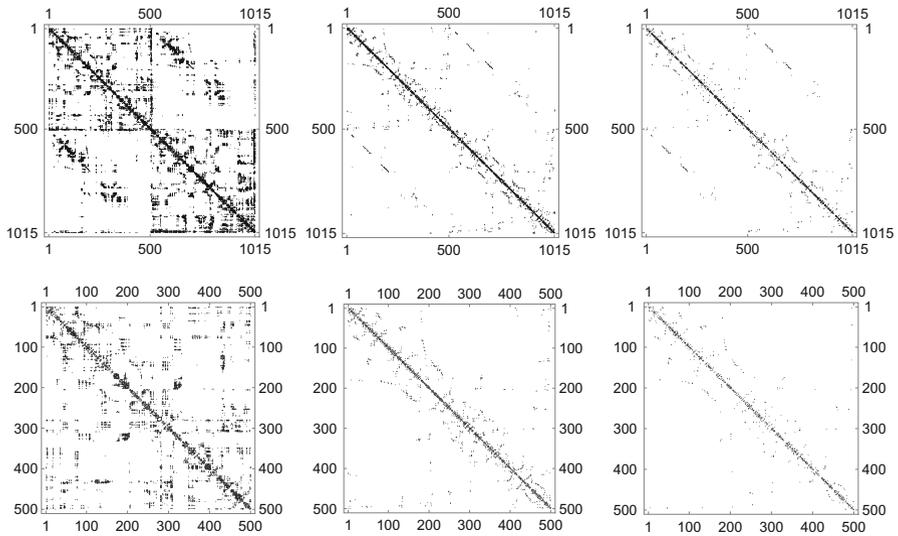


Fig. 4 Array plots of the adjacency matrix of a 1015 node (Scale 5) structural brain network, the Greedy Navigational Core (GNC) network, and that part of the GNC which is included in the brain network. Complete networks are in the first row; the upper left part of the networks (500 nodes) are in the second row

to brain networks. This is illustrated by Fig. 4 where one can observe that in a Scale 5 GNC network and also in that part of the GNC which is included in the brain similar patterns can be identified as in the original brain network. Even the false-positive links in GNC (those links which are not included in the brain) form such arrangements which seems to be a smooth “continuation” of the original brain patterns. This may relate to the fact that the false-positive links in a GNC (corresponding to a given brain network) are likely present (with probability well above 0.9) in at least one of the remaining 39 brain networks.

As the results showed, the increased resolution of brain parcellation does not necessarily imply the decrease of GNC precision. Note that the highest inclusion ratios are in Scale 2 and the coarser Scale 1 and finer Scale 3 provide somewhat lower precisions. Significantly lower (but still high and consistent) precisions can be observed in Scales 4 and 5. One can speculate that in these finer brain parcellations, the 3D Euclidean space is not as suitable for GNC induction as in lower scales. One possible reason for this is that the brain cortex is highly folded and this may cause in higher resolutions that the lengths of curved fiber paths between brain parcels are less correlated to the Euclidean distances.

4 Methods

Data The dataset used in our investigations contains 40 healthy human subjects who underwent an MRI measurement procedure where diffusion spectrum imaging (DSI) data were obtained for each subject. The DSI data was processed according to the methods described in [11], resulting in 40 weighted, undirected structural connectivity maps comprising 83, 129, 233, 463, and 1015 nodes in five different scales, respectively. Each node represents a parcel of cortical or subcortical gray matter, and connections represent white matter streamlines connecting a pair of brain regions. Connection weights determine the average density of white matter streamlines and here consider connections with density above 10^{-8} , resulting structural networks with an average of 1119, 1976, 3799, 7246, and 14,254 connections per subject.

Greedy Navigational Cores The Greedy Navigational Core is generated from an empty network, using only the coordinates of nodes as input parameters. GNC can be considered as the solution of a constrained optimization problem, in which the goal is to reach 100% greedy navigability with setting up minimum number of links between the nodes. This hard discrete optimization task can be traced back to the well-known minimum set cover problem, for solving that computationally efficient heuristics is available [13]. Searching the GNC can also be formulated as a special network formation game (called network navigation game) in which the selfish players have strategies to set up links according to a payoff function in order to reach each other with greedy routes. An important property of GNC is that it is the Nash equilibrium of the network navigation game.

Acknowledgments Project nos. 123957, 129589, and 124171 have been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the FK_17, KH_18, and K_17 funding schemes, respectively. Z. Heszbarger and A. Gulyas have been supported by the Janos Bolyai Fellowship of the Hungarian Academy of Sciences and by the UNKP-19-4 New National Excellence Program of the Ministry of Human Capacities.

References

1. A. Gulyás, J.J. Bíró, A. Kőrösi, G. Rétvári, D. Krioukov, Navigable networks as nash equilibria of navigation games. *Nat. Commun.* **6**, 7651 (2015)
2. O. Sporns, *Networks of the Brain* (MIT Press, 2010)
3. A. Fornito, A. Zalesky, E. Bullmore, *Fundamentals of Brain Network Analysis* (Academic Press, 2016)
4. S.N. Sotiropoulos, A. Zalesky, Building connectomes using diffusion MRI: why, how and but. *NMR Biomed.* **32**(4), e3752 (2019)
5. D.S. Bassett, E.T. Bullmore, Small-world brain networks revisited. *The Neuroscientist* **23**(5), 499–516 (2017)

6. C. Seguin, M.P. Van Den Heuvel, A. Zalesky, Navigation of brain networks. *Proc. Natl. Acad. Sci.* **115**(24), 6297–6302 (2018)
7. A. Allard, M.Á. Serrano, Navigable maps of structural brain networks across species. *PLoS Comput. Biol.* **16**(2), e1007584 (2020)
8. I. Pappas, M.M. Craig, D.K. Menon, E.A. Stamatakis, Structural optimality and neurogenetic expression mediate functional dynamics in the human brain. *Hum. Brain Mapp.* **41**(8), 2229–2243 (2020)
9. D. Zhou, C.W. Lynn, Z. Cui, R. Ciric, G.L. Baum, T.M. Moore, D.R. Roalf, J.A. Detre, R.C. Gur, R.E. Gur et al., Efficient coding in the economics of human brain connectomics. *arXiv preprint arXiv:2001.05078* (2020)
10. A. Muscoloni, C.V. Cannistraci, Navigability evaluation of complex networks by greedy routing efficiency. *Proc. Natl. Acad. Sci.* **116**(5), 1468–1469 (2019)
11. R.F. Betzel, A. Avena-Koenigsberger, J. Goñi, Y. He, M.A. De Reus, A. Griffa, P.E. Vértés, B. Mišić, J.-P. Thiran, P. Hagmann et al., Generative models of the human connectome. *Neuroimage* **124**, 1054–1064 (2016)
12. M.A. de Reus, M.P. van den Heuvel, Estimating false positives and negatives in brain networks. *Neuroimage* **70**, 402–409 (2013)
13. U. Feige, L. Lovász, P. Tetali, Approximating min sum set cover. *Algorithmica* **40**(4), 219–234 (2004)

A Multicommodity Flow Formulation and Edge Exchange Heuristic Embedded in Cross Decomposition for Solving Capacitated Minimum Spanning Tree Problem



Han-Suk Sohn and Dennis Bricker

1 Introduction

The minimum spanning tree problem is a fundamental problem in the design of centralized data communication networks. Many variations of the minimum spanning tree problem occur in the field of communication networks and computer networks. One of them is the capacitated minimum spanning tree (CMST) problem, which is concerned with finding a spanning tree of minimum total edge weight, such that each branch from the center has a sum of demands that does not exceed a given capacity. The CMST problem has many applications in network design and centralized telecommunications. For example, the terminal node can be a central power plant, a central communication controller, or a central computer shared by many users; then, the capacity limits the maximum flow on each branch or the amount of traffic each port at the center can handle [1]. The problem of finding a CMST also arises in many other areas such as transportation, communication, plumbing, sewage, etc. Many authors have proposed mathematical formulations for the CMST [2–10]. However, due to its NP-hard nature (see [11]), the solution of the CMST is usually very time consuming even for moderate size instances [12]. Van Roy developed the cross decomposition algorithm, which unifies Benders' decomposition and Lagrangian relaxation into a single framework [13] and it has been successfully implemented in many areas [13–18]. In this chapter, to achieve

H.-S. Sohn (✉)

Industrial Engineering, New Mexico State University, Las Cruces, NM, USA
e-mail: hsohn@nmsu.edu

D. Bricker

Industrial Engineering, University of Iowa, Iowa City, IA, USA
e-mail: dbricker@uiowa.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_26

347

effectiveness in solving the CMST, we present a new mathematical formulation and propose a cross decomposition-based algorithm.

2 Single-Commodity Flow Formulation

Consider the following single-commodity flow formulation of the capacitated minimum spanning tree (CMST) problem used:

(Problem 1)

$$\text{Minimize } \sum_{i=2}^n \sum_{\substack{j=1 \\ j \neq i}}^n c_{ij} X_{ij} \tag{1}$$

$$\text{subject to } \sum_{\substack{j=1 \\ j \neq i}}^n X_{ij} = 1 \quad i = 2, \dots, n \tag{2}$$

$$\sum_{\substack{j=1 \\ j \neq i}}^n y_{ij} - \sum_{\substack{j=2 \\ j \neq i}}^n y_{ji} = 1 \quad i = 2, \dots, n \tag{3}$$

$$y_{ij} \leq K X_{ij} \quad i = 2, \dots, n; j = 1, \dots, n \tag{4}$$

$$y_{ij} \geq 0 \text{ and } X_{ij} = 0 \text{ or } 1 \quad \forall i, j \tag{5}$$

where X_{ij} is 1 if arc (i, j) is included in the minimum spanning tree, and 0 otherwise, y_{ij} is the flow on the arc (i, j) , and K is the capacity restriction limit on the flow on any link. Notationally, if (\bullet) is an optimization problem, we let $v(\bullet)$ be its optimal solution value, $\bar{v}(\bullet)$ its incumbent objective value and $F(\bullet)$ its feasible region.

3 Cross Decomposition to (Problem 1)

Dual Decomposition (Lagrangian Relaxation) Dual decomposition of (Problem 1) is outlined below. The formal Lagrangian dual of (Problem 1) relative to the constraints $y_{ij} \leq KX_{ij}$ is given by (D)

$$\begin{aligned}
 (D) \text{ Maximize}_{u \geq 0} & \left. \begin{array}{l}
 \text{Minimize}_{x \in Z} \sum_{i=2}^n \sum_{\substack{j=1 \\ j \neq i}}^n c_{ij} X_{ij} + \sum_{i=2}^n \sum_{\substack{j=1 \\ j \neq i}}^n u_{ij} (y_{ij} - KX_{ij}) \\
 y \geq 0 \\
 \text{subject to} \sum_{\substack{j=1 \\ j \neq i}}^n X_{ij} = 1, \quad i = 2, \dots, n \\
 \sum_{\substack{j=1 \\ j \neq i}}^n y_{ij} - \sum_{\substack{j=2 \\ j \neq i}}^n y_{ji} = 1, \quad i = 2, \dots, n
 \end{array} \right| \\
 & = \text{Maximize}_{u \geq 0} v(\text{DS}(u))
 \end{aligned}$$

that is, the inner minimization problem in (D) is defined as the dual subproblem and the dual master problem can be written as:

$$(MAD) \left. \begin{array}{l}
 \text{Maximize}_{u \geq 0, u_0 \in R} u_0 \\
 \text{subject to} \quad u_0 \leq \left\{ \sum_{i=2}^n \sum_{\substack{j=1 \\ j \neq i}}^n c_{ij} X_{ij}^d + \sum_{i=2}^n \sum_{\substack{j=1 \\ j \neq i}}^n u_{ij} (y_{ij} - KX_{ij}^d) \right\}, \quad d \in D_{DA}
 \end{array} \right|$$

$\{x^d, d \in D_{DA}\}$ is the set of extreme points of $F(\text{DS}(u))$.

Primal (Benders') Decomposition Primal decomposition of (Problem 1) is implemented as follows:

$$\begin{aligned}
 (P) \text{ Minimize } & \left. \begin{array}{l} \text{Minimize}_{y \geq 0} \sum_{i=2}^n \sum_{\substack{j=1 \\ j \neq i}}^n 0 y_{ij} + \sum_{i=2}^n \sum_{\substack{j=1 \\ j \neq i}}^n c_{ij} X_{ij} \\ \text{subject to} \sum_{j=1}^n y_{ij} - \sum_{\substack{j=2 \\ j \neq i}}^n y_{ji} = 1, i = 2, \dots, N \\ y_{ij} \leq K X_{ij}, i = 2, \dots, N; j = 1, \dots, n \end{array} \right| \\
 & = \text{Minimize}_{x \in Z} v(\text{PS}(x))
 \end{aligned}$$

For any fixed value of x , the inner minimization problem of (P) is a linear program which is called the primal or Benders' subproblem. By dualizing this linear program $(\text{PS}(x))$ we may rewrite (P) as:

$$\begin{aligned}
 (P) \text{ Minimize } & \left. \begin{array}{l} \text{Maximize}_{v \geq 0, u \geq 0} \sum_{i=2}^n v_i - \sum_{i=2}^n \sum_{\substack{j=1 \\ j \neq i}}^n K X_{ij} u_{ij} + \sum_{i=2}^n \sum_{\substack{j=1 \\ j \neq i}}^n c_{ij} X_{ij} \\ \text{subject to} v_i - v_j - u_{ij} \leq 0, i = 2, \dots, n; j = 2, \dots, n \\ v_i - u_{ij} \leq 0, i = 2, \dots, n \end{array} \right| \\
 & = \text{Minimize}_{x \in Z} v(\text{PS}_D(x))
 \end{aligned}$$

Primal master problem can be written as:

$$\begin{aligned}
 (\text{MA}_P) \text{ Minimize } & \left. \begin{array}{l} x_0 \\ x \in Z, x_0 \in R \\ \text{subject to } x_0 \geq \left\{ \sum_{i=2}^n v_i^d - \sum_{i=2}^n \sum_{\substack{j=1 \\ j \neq i}}^n K X_{ij} u_{ij}^d + \sum_{i=2}^n \sum_{\substack{j=1 \\ j \neq i}}^n c_{ij} X_{ij} \right\}, d \in \text{DPA} \\ \sum_{\substack{j=1 \\ j \neq i}}^n X_{ij} = 1, i = 2, \dots, n \end{array} \right|
 \end{aligned}$$

In the above, $\{v^d \ \& \ u^d, \ d \in \text{DPA}\}$ is the set of extreme points of $F(\text{PS}_D(x))$. The constraints of the primal master problem are called the Benders' cuts or primal cuts and are generated by the dual solutions of the primal subproblem. Note that if $(\text{PS}_D(x))$ is unbounded, in other words, if u goes to infinity with KXu finite (i.e.,

(PS(x)) is infeasible), then add the regularity constraint (i.e., $\sum_i v_i \leq M$, where M is very large number) to the problem and solve (PS_D(x)) again.

4 Multicommodity Flow Formulation

Developing effective cross decomposition algorithms for the CMST problem requires efficient solution of the dual and primal subproblems. The dual subproblem requires solving shortest path problems and a minimum spanning tree problem, for which efficient algorithms exist. Solution of the primal subproblem, a network flow problem, is also easily accomplished—at least the computation of the primal (flow) variables. Network flow problems are known to be degenerate, however, which means that multiple dual solutions may exist. Dealing with this degeneracy in the most effective way is the key to the success of a cross decomposition algorithm. In this chapter, therefore, a multicommodity flow formulation for CMST problem is proposed as below:

The CMST problems appear to be very amenable to dual decomposition, but less so to primal (a.k.a. Benders') decomposition, when they are formulated as mixed-integer LP problems: x_{ij} has a value of 1 if link (i, j) is in the tree, 0 otherwise. y_{ij} is flow of (a *single*) commodity in link (i, j) where each terminal node is the source of one unit and the central node has a demand for $(n - 1)$ units. Here, the *complicating* constraint (e.g., the capacity constraint on the links, which is a constraint on the y variables, or the degree constraint on the nodes, which is a constraint on the x variables) is relaxed. *Primal decomposition*, however, is more problematic, if the cost of the links is assigned only to the x variables. This would mean (if we use x as the “complicating variables” which are tentatively fixed in Benders' decomposition) that the primal subproblem has *no objective function*—it is either *feasible* (in which case the dual variables required in cross decomposition are all zero) or it is *infeasible* (in which case the dual variables reflect the penalties assigned to the artificial variables, which are independent of the data). This does not provide much information (in the way of dual variables) for the dual subproblems. What is needed is a formulation in which the cost of the links is associated with the y variables. Consider then the following multicommodity flow version of the problem, where y_{ij}^k is the flow of commodity k in link (i, j) , where commodity k originates at node k and is destined for the central node, node #1 and, as is usual, x_{ij} takes a value of 1 if link (i, j) is included in the tree, and 0 otherwise. Now the cost c_{ij} of link (i, j) can be associated with the *first* link in the path carrying commodity i to the central node, that is, the cost of including link (i, j) in the tree is given by the term $c_{ij} y_{ij}^i$. Only the flow in the first link out of node k has an associated cost—the remaining flows are “free” of cost). The formulation will then be:

(Problem 2)

$$\text{Minimize } \sum_k \sum_j c_{kj} y_{kj}^k \tag{6}$$

$$\text{subject to } \left(\sum_j y_{ij}^k \right) - \left(\sum_j y_{ji}^k \right) = \begin{cases} +1 & \text{if } i = k \\ 0 & \text{if } i \neq 1 \text{ or } k \\ -1 & \text{if } i = 1 \end{cases} \quad \text{for all } i \& k \tag{7}$$

$$x \in X \equiv \text{family of spanning trees of the network} \tag{8}$$

$$y_{ij}^k \leq x_{ij} \quad \forall (i, j) \in A \quad \& \quad k \tag{9}$$

$$x_{ij}, y_{ij}^k \in \{0, 1\} \quad \forall i, j, k \tag{10}$$

$$\sum_k y_{i1}^k \leq K x_{i1} \quad \forall i = 2, \dots, n \tag{11}$$

where K is the capacity of the links. The objective function (6) to be minimized is the total cost of the communication arcs selected for the tree. The first constraint set (7) implies the *conservation of flow equations* for commodity k , that is, one unit originates at node k , destined for node 1. The constraint (8) implies the spanning tree constraints for x . Note that these constraints are considered to be implicitly expressed, rather than requiring constraints to eliminate cycles. The third constraint (9) links the x & y variables, which are to be relaxed in the dual decomposition. The fourth constraint (10) implies that the flows on all arcs are nonnegative, and an arc is either used or is not used. The last constraint (11) puts an explicit upper limit on the flow.

5 Cross Decomposition to (Problem 2)

Dual Subproblem Dual subproblem of (Problem 2) is as follows:

$$\text{Minimize } \sum_k \sum_j c_{kj} y_{kj}^k + \sum_i \sum_j \sum_k \lambda_{ij}^k (y_{ij}^k - x_{ij}) + \sum_i \mu_i \left(\sum_k y_{i1}^k - K x_{i1} \right) \tag{6'}$$

subject to: (7), (8), and (10)

The objective function of the Lagrangian subproblem may be restated as:

$$\text{Minimize } \sum_i \sum_j \left(c_{ij} \delta_{ik} + \lambda_{ij}^k + \mu_i \right) y_{ij}^k - \sum_i \sum_j \left(\sum_k \lambda_{ij}^k + K \mu_i \right) x_{ij} \tag{6''}$$

where δ_{ik} is the Kronecker delta (i.e., $d_{ik} = 1$, if $i = k$, and 0 otherwise). This problem separates into $(n - 1)$ *shortest-path* problems in the variables y_{ij}^k (from each terminal node to the central node) and a *minimum spanning tree* problem in the variables x_{ij} .

Primal Subproblem In the primal subproblem, the spanning tree x is fixed, and a multicommodity minimum cost network flow problem (with upper bounds on both individual commodity flows and, in the case of arcs incident to the central node, on the aggregate flows) is to be solved. Unlike the formulations presented in Sect. 2, the objective function is not the constant zero, but is the actual cost of the spanning tree. Hence, the primal subproblem does not merely check for feasibility as before, but should generate more useful dual variables to guide the search for the optimal spanning tree.

6 Edge Exchange Heuristic

When applying cross decomposition to the capacitated minimum spanning tree problem, the solution of the dual subproblem (i.e., the Lagrangian relaxation) yields values of X , which specifies the links to be included in the tree. This tree will generally violate one or more of the link capacity constraints. A Lagrangian heuristic algorithm is an algorithm which aims to modify the infeasible solution of the Lagrangian relaxation in order to obtain a “good” feasible solution, and an upper bound on the optimum. This heuristic algorithm could be applied at every iteration, or regularly but less frequently, or when the infeasibility is less than some threshold. The proposed heuristic algorithm may be considered as an “edge exchange” algorithm. For convenience, notations used in the heuristic algorithm are summarized as below:

N Set of nodes, excluding the “central” node c

d_{ij} Length of edge (i, j)

K Capacity to be imposed. For simplicity of explanation, it is assumed that the same value is used for all edges, although the algorithm to be described can be generalized

X_{ij} Binary variables obtained from the Lagrangian (dual) subproblem, with 1 indicating that edge (i, j) is to be included in the tree. For simplicity of

explanation, it is assumed that X to be fixed previously by the dual subproblem rather than variable

T Spanning tree specified by X

$R \subseteq N$ Set of nodes for which $X_{ck} = 1$, that is, the nodes connected directly to the central node

$T_k \subseteq T$ Subtree which is rooted at node k , where $k \in R$

$\hat{N} \subseteq N$ is the set of nodes with degree 1 in the tree specified by X

\widetilde{N} Set of nodes with degree greater than 1, except for those nodes k such that $\{k\} = T_k$, that is, the subtree consists of a single node. These nodes can be expressed as $(N \sim \hat{N}) \cup (R \cap \hat{N})$

β_k Excess number of nodes in tree T_k if positive, or the slack in the capacity constraint of edge (c, k) if negative. These nodes can be expressed as $K - |T_k|$

Z_{ik} Binary decision variable, with value 1 indicating that node i is to be detached from its subtree and reattached to T_k , where $i \in \hat{N}$ and $k \in R$

δ_{ik} Minimum amount of increase in cost if node i is detached from its subtree and re-attached to T_k , where $i \in \hat{N}$ and $k \in R$. That is,

$$\delta_{ik} = \text{Min}_{j \in N \cap T_k} \{d_{ij}\} - \sum_{q \in N} d_{qi} X_{qi}$$

To modify X in order to obtain a feasible spanning tree, we solve the following problem.

$$\text{Minimize } \sum_{i \in \hat{N}} \sum_{T_k \subseteq T} \delta_{ik} Z_{ik} \tag{12}$$

$$\text{subject to } \sum_{i \in T_k} \sum_{j \in R} Z_{ij} - \sum_{j \in R} \sum_{i \in T_k} Z_{ji} \begin{cases} = \beta_k & \text{if } \beta_k \geq 0 \\ \geq \beta_k & \text{if } \beta_k < 0 \end{cases} \text{ for all } k \in R \tag{13}$$

$$0 \leq Z_{ij} \leq 1 \quad \text{for } i \in \hat{N}, j \in R \tag{14}$$

The problem can be viewed as akin to a single-commodity network flow problem. Constraint (13) imposes “conservation of flow” on the subtree T_k , that is, a restriction of “flow out – flow in” which would result in a feasible spanning tree. Note that only nodes of degree 1 are detached from a subtree and these can be reattached only to nodes with degree greater than 1, except possibly for singleton nodes, that is, nodes of degree 1 attached directly to the central node, although improved solutions might require other types of modifications. This restriction is necessary to prevent a node being reattached to another subtree via a node which itself is being detached from its subtree and reattached elsewhere. Unlike previous

edge exchange algorithm, this new algorithm is implemented by solving a linear program (LP), and is not restricted to “one-for-one” exchanges. For the purpose of illustration, let us consider a spanning tree in Fig. 1, where c is the central node, $N = \{1, \dots, 13\}$, $\hat{N} = \{1, 2, 3, 7, 9, 10, 11, 13\}$ and a given capacity $U = 3$. Then the set of subtrees is $R = \{5, 6, 8, 10, 12\}$, where $T_5 = \{2, 3, 5\}$ with $\beta_5 = 0$, $T_6 = \{6, 9\}$ with $\beta_6 = -1$, $T_8 = \{1, 4, 7, 8\}$ with $\beta_8 = +1$, $T_{10} = \{10\}$ with $\beta_{10} = -2$, and $T_{12} = \{11, 12, 13\}$ with $\beta_{12} = 0$. The set of nodes that are candidates for attachments are $\hat{N} = \{4, 5, 6, 8, 10, 12\}$. The costs of modifications are, for example, $\delta_{1,5} = d_{1,2} - d_{4,1}$ (the net change in length of the tree if node 1 is detached from the tree T_8 and reattached to tree T_5 via node 2, the nearest node T_5 having degree >1).

Suppose that the solution of the LP is $Z_{1,5} = Z_{3,6} = 1$. Then the cardinality of subtree T_8 has been reduced to $U = 3$, the cardinality of subtree T_5 has a net change of zero, and the cardinality of subtree T_6 has increased by 1, as shown in Fig. 2. After the dual subproblem has been solved and an infeasible spanning tree X has resulted, the subtrees of X must be identified. This can be accomplished by solving the primal subproblem with the fixed values of X , but with no capacity restrictions. Let Y_{ij}^k be the solution. Then $R = \{k \in N \mid X_{ck} = 1\}$ and $j \in T_k$ if $Y_{ck}^j = 1$. The LP to implement the heuristic algorithm may be defined and solved. Then the primal subproblem may be solved by changing the upper bound constraints on those edges for which $X_{ck} = 1$.

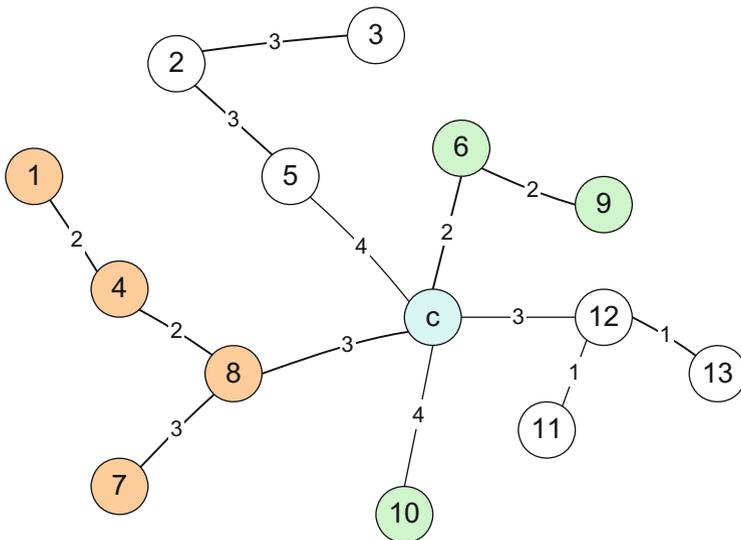


Fig. 1 Infeasible spanning tree

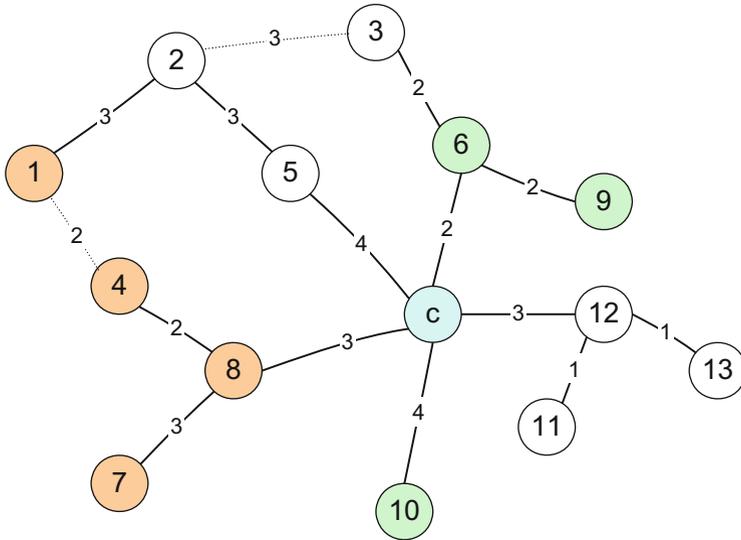


Fig. 2 Modifications $Z_{1,5} = Z_{3,6} = 1$

7 Computational Study

The proposed algorithm was programmed in Mosel language for FICO Xpress and run on an Intel(R) Xeon(R) CPU 2.67 GHz with 12 GB of Ram. In order to compare the performance of the proposed algorithm with the performance of another algorithm reported in the literature (i.e., the Lagrangian dual algorithms of Gouveia [3]), “CRD40#” data sets that are a class of 40-node (excluding central node) symmetric instances from OR-Library [19] have been used. Note that, in the Lagrangian relaxation (LR) algorithm, a subgradient method was used to search for the Lagrangian multiplier. Based upon prior computational experience and common usage, a scalar for the step size factor in subgradient optimization for the Lagrangian dual algorithm is determined by starting with an initial value of 2 and reducing it by a factor of 1/1.1 whenever the dual subproblem solution has failed to increase within 10 iterations.

Tables 1, 2, and 3 summarize the computational results for a set of 40-node problems. The tables include percentage gaps between the best upper bound and the best lower bound within 500 iterations or within 15 hours of computational time by the Lagrangian dual algorithm with edge exchange heuristic algorithm (LR with EXH), by the ordinary cross decomposition algorithm with edge exchange heuristic (XD with EXH), and by the ordinary cross decomposition algorithm without edge exchange heuristic (XD w/o EXH). Also, BUB and BLB represent the best upper bound and the best lower bound, respectively. Three different values for link capacity (3, 5, and 10) were used for each of 10 test problems (CRD4001 through CRD40010) were solved. For the case of link capacity 3, the computational

Table 1 Comparison for a set of 40-node problem (link capacity = 3)

Problem	LR with EXH			XD w/o EXH			XD with EXH		
			Gap			Gap			Gap
ID	BUB	BLB	(%)	BUB	BLB	(%)	BUB	BLB	(%)
crd4001	752	593	21.20 ^a	853	689	19.23 ^a	750	689	8.13 ^a
crd4002	726	574	20.92 ^a	810	665	17.90 ^a	725	665	8.28 ^a
crd4003	753	602	20.08 ^a	849	652	23.20 ^a	753	652	13.41 ^a
crd4004	791	634	19.88 ^a	910	689	24.29 ^a	785	689	12.23 ^a
crd4005	771	552	28.44 ^a	922	673	27.01 ^a	771	673	12.71 ^a
crd4006	758	528	30.37 ^a	889	666	25.08 ^a	747	666	10.84 ^a
crd4007	771	550	28.64 ^a	937	695	25.83 ^a	760	695	8.55 ^a
crd4008	732	511	30.20 ^a	856	632	26.17 ^a	732	632	13.66 ^a
crd4009	795	624	21.56 ^a	828	680	17.87 ^a	789	680	13.81 ^a
crd40010	798	606	24.08 ^a	863	698	19.12 ^a	798	698	12.53 ^a

^aTerminated due to the computational time limitation (15 hours)

Table 2 Comparison for a set of 40-node problem (link capacity = 5)

Problem	LR with EXH			XD w/o EXH			XD with EXH		
			Gap			Gap			Gap
ID	BUB	BLB	(%)	BUB	BLB	(%)	BUB	BLB	(%)
crd4001	589	494	16.07 ^a	613	556	9.23 ^a	586	556	5.08 ^a
crd4002	587	498	15.23 ^a	626	554	11.55 ^a	578	554	4.13 ^a
crd4003	577	503	12.75 ^a	605	564	6.76 ^a	577	564	2.25 ^a
crd4004	620	534	13.85 ^a	648	585	9.76 ^a	617	585	5.23 ^a
crd4005	600	519	13.44 ^a	633	587	7.28 ^a	600	587	2.12 ^a
crd4006	607	520	14.38 ^a	617	555	10.01 ^a	590	555	5.87 ^a
crd4007	609	495	18.76 ^a	643	587	8.72 ^a	609	587	3.55 ^a
crd4008	553	496	10.28 ^a	569	544	4.39 ^a	553	544	1.63 ^a
crd4009	599	498	16.85 ^a	619	556	10.29 ^a	599	556	7.22 ^a
crd40010	605	501	17.25 ^a	659	574	12.83 ^a	600	574	4.27 ^a

^aTerminated due to the computational time limitation (15 hours)

times from all three approaches are disappointing. It is noted that a relatively small number of iterations, in most cases, less than 50, have been performed within 15 hours of computational time. This is because Xpress did not use the more specialized algorithm for each subproblem. Note that the dual subproblem can be separated into the minimum spanning tree problem and the shortest-path problem, which can be solved using methods specifically tailored for them. The network structure of the primal subproblem can be better exploited by the solver CPLEX, whereas Xpress is a more general purpose linear programming solver which uses the ordinary simplex algorithm even for a pure network problem. Therefore, the computational performance could be improved by using CPLEX for solving the primal subproblem. The solution of the dual subproblem is taking much more time than that of the primal subproblem. However, the improvement obtainable by using

Table 3 Comparison for a set of 40-node problem (link capacity = 10)

Problem	LR with EXH			XD w/o EXH			XD with EXH		
			Gap			Gap			Gap
ID	BUB	BLB	(%)	BUB	BLB	(%)	BUB	BLB	(%)
crd4001	502	463	7.80 ^a	519	494	4.86 ^a	498	494	0.80 ^a
crd4002	490	462	5.73 ^b	507	489	3.55 ^a	490	489	0.20 ^a
crd4003	500	484	3.27 ^b	500	500	0.00 ^b	500	500	0.00 ^b
crd4004	520	494	4.97 ^a	512	501	2.15 ^a	512	501	2.15 ^a
crd4005	504	478	5.11 ^b	504	504	0.00 ^b	504	504	0.00 ^b
crd4006	498	474	4.87 ^b	498	490	1.61 ^b	498	495	0.60 ^b
crd4007	514	477	7.16 ^b	508	487	4.13 ^b	508	502	1.18 ^b
crd4008	485	448	7.55 ^b	485	485	0.00 ^b	485	485	0.00 ^b
crd4009	528	486	7.89 ^a	516	500	3.10 ^a	516	508	1.55 ^a
crd40010	532	487	8.52 ^a	517	492	4.84 ^a	517	507	1.93 ^a

^aTerminated due to the computational time limitation (15 hours)

^bTerminated due to the iteration number limitation (500 iterations)

CPLEX for solving the primal subproblem is probably not significant. On the other hand, the solution of the dual subproblem could be accomplished more efficiently, using specialized algorithms for the minimum (uncapacitated) spanning tree and shortest path problems. Furthermore, computational time for the occasional solution of the primal (Benders’) master problem, an integer programming problem with little structure, could be significantly reduced by not requiring that it be solved to optimality, but by terminating with a suboptimal solution with the property that its value is less than the best upper bound. Even though the computational times are disappointing, the performances of all three approaches are significantly improved as the link capacity increases. It is also noted that the edge exchange heuristic strengthens the upper bound in the cross decomposition algorithm. Figures 3, 4, and 5 shows that convergence of both cross decomposition and Lagrangian dual algorithm are improved as the link capacity increases for the CRD4003 instance, for example. From these figures, we note that the cross decomposition algorithm converges more rapidly than the Lagrangian dual algorithm, although more iterations have been performed in Lagrangian dual algorithms within a specified time. Looking at the average behavior over the CRD data sets, the Lagrangian dual algorithm is not comparable to the performance of the cross decomposition algorithm in terms of the ratio of the best upper bound and the best lower bound.

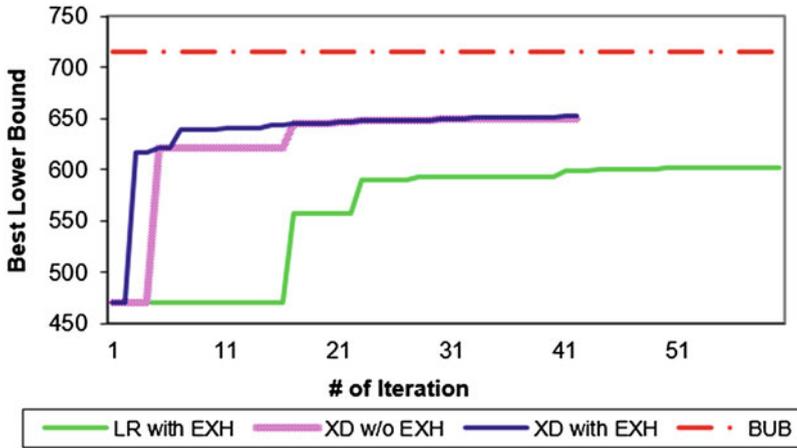


Fig. 3 Lower bounds of three approaches (CMST with link capacity 3 for CRD4003 instance)

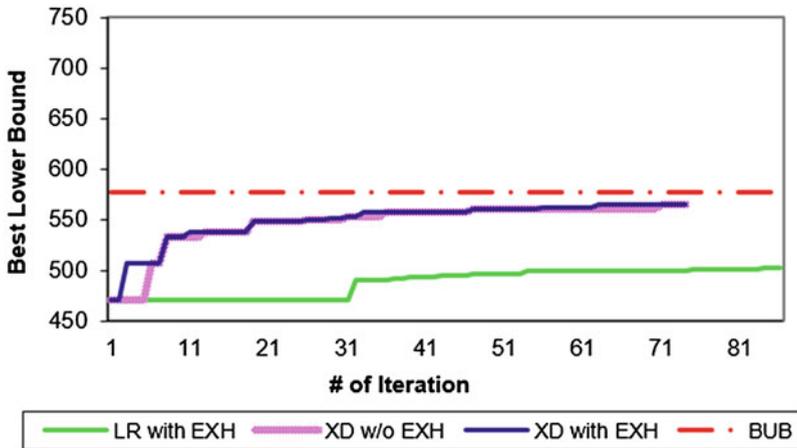


Fig. 4 Lower bounds of three approaches (CMST with link capacity 5 for CRD4003 instance)

8 Conclusions

In this chapter, an implementation of Van Roy’s cross decomposition algorithm for CMST problem has been described and the corresponding computational results have been analyzed. Although the excessive computational times are disappointing, using the new mathematical formulation based on multicommodity flow together with the proposed edge exchange heuristic algorithm does improve the performance of the cross decomposition algorithm significantly. It is also shown to the proposed algorithm converges more rapidly than the Lagrangian dual algorithm for the CMST problem. The use of the new formulation and the proposed algorithm which take

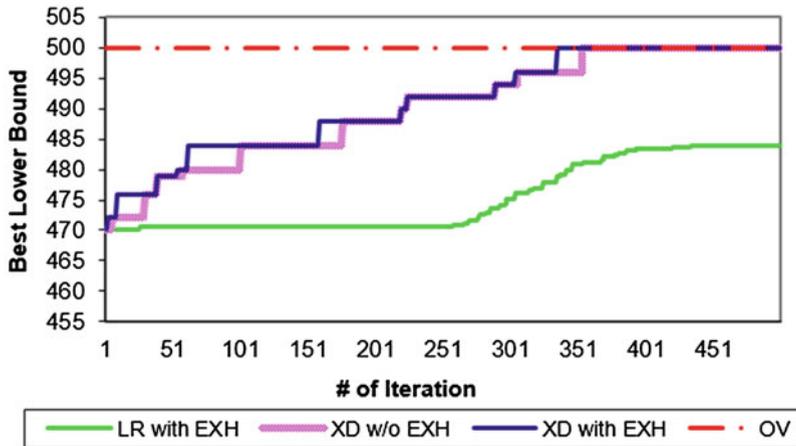


Fig. 5 Lower bounds of three approaches (CMST with link capacity 10 for CRD4003 instance)

better advantage of the problem structure, especially that of the dual subproblem, provides a large potential for improvement.

References

1. K. Malik, G. Yu, A branch-and-bound algorithm for the capacitated minimum spanning tree problem. *Networks* **23**, 525–532 (1993)
2. J.R. Araque, L.A. Hall, T.L. Magnanti, *Capacitated trees, capacitated routing, and associated polyhedra. Technical report SOR-90-12, Program in Statistics and Operations Research* (Princeton University, Princeton, 1990)
3. L. Gouveia, A comparison of directed formulations for the capacitated minimal spanning tree problem. *Telecommun. Syst.* **1**, 51–76 (1993)
4. L. Gouveia, A $2n$ -constraint formulation for the capacitated minimal spanning tree problem. *Oper. Res.* **43**, 130–141 (1995)
5. L. Gouveia, Multicommodity flow models for spanning trees with hop constraints. *Eur. J. Oper. Res.* **95**, 178–190 (1996)
6. L. Gouveia, L. Hall, Multistars and directed flow formulations. *Networks* **40**, 188–201 (2002)
7. L. Gouveia, M. Lopes, The capacitated minimum spanning tree problem: On improved multistar constraints. *Eur. J. Oper. Res.* **160**, 47–62 (2005)
8. L. Gouveia, P. Martins, A hierarchy of hop-indexed models for the capacitated minimal spanning tree problem. *Networks* **35**, 1–16 (2000)
9. L. Gouveia, P. Martins, The capacitated minimum spanning tree problem: Revisiting hop-indexed formulations. *Comput. Oper. Res.* **32**, 2435–2452 (2005)
10. L. Hall, Experience with a cutting plane algorithm for the capacitated spanning tree problem. *INFORMS J. Comput.* **8**, 219–234 (1996)
11. C. Papadimitriou, The complexity of the capacitated tree problem. *Networks* **8**, 217–230 (1978)

12. E. Ruiz, M. Albareda-Sambola, E. Fernandez, M. Resende, A biased random-key genetic algorithm for the capacitated minimum spanning tree problem. *Comput. Oper. Res.* **57**, 95–108 (2015)
13. T.J. Van Roy, Cross decomposition for mixed integer programming. *Math. Program.* **25**, 46–63 (1983)
14. N. Deeb, S. Shahidehpour, Cross decomposition for multi-area optimal reactive power planning. *IEEE Trans. Power Syst.* **8**(4), 1539–1544 (1993)
15. K. Holmberg, K. Jornsten, Cross decomposition applied to the stochastic transportation problem. *Eur. J. Oper. Res.* **17**(3), 361–368 (1984)
16. C. Lee, A cross decomposition algorithm for a multiproduct-multitype facility location problem. *Comput. Oper. Res.* **20**(5), 527–540 (1993)
17. H. Sohn, D.L. Bricker, Cross decomposition of the degree-constrained minimum spanning tree problem. *J. Syst. Cybern. Inf.* **5**(1), 31–34 (2007)
18. C. Yoo, D. Tcha, A cross decomposition procedure for the facility location problem with a choice of facility type. *Comput. Ind. Eng.* **10**(4), 283–290 (1986)
19. J.E. Beasley, OR-library: Distributing test problems by electronic mail. *J. Oper. Res. Soc.* **41**(11), 1069–1072 (1990)

Elemental Analysis of Oil Paints



Shijun Tang, Rosemarie C. Chinni, Amber Malloy, and Megan Olsson

1 Introduction

Art works (e.g., painting works) have a long history and significant cultural value. Also, pigments, techniques, mediums, and historical documents provide substantial information about an artwork and its creator. However, there are shortcomings that experts in the field cannot overcome without technological assistance. Experts cannot precisely find pigments, techniques, mediums, historical and creator's information only via their eyes.

Even pigments made of the same substance can vary according to different ratios of components. The human eyes cannot make such differentiations. Besides this limitation, an art historian or conservator cannot see beneath the surface of the artwork. Should any valuable information about the artist's style and process be hidden there, experts are physically unable to observe it. Scientific analysis is essential to make these subtle distinctions and offer supplementary information [1].

Before the eighteenth century, a basic palette consisted of 15 pigments, many of which were difficult to use because of low opacity, weak hues, and chemical instability. Along with the 40 new elements added to the Periodic Table and two other developments followed which enabled the artist to paint more freely. Catalyzed by the scientific developments from the Enlightenment, the manufacture of paint changed in a fundamental way. Another growing concern was storage and transportation [1]. All these inspired a new wave of artists to emerge—the impressionists. These artists were able to leave the confines of the studios and observe natural lighting. They used fewer pigments than their predecessors because the synthesized materials were so pronounced. The impressionists also used fewer

S. Tang (✉) · R. C. Chinni · A. Malloy · M. Olsson
Department of Science and mathematics, Alvernia University, Reading, PA, USA
e-mail: shijun.tang@alvernia.edu

layers, which greatly reduced the drying time and allowed them to capture the fleeting “impressions” of scenery for which they are known.

The composition of paints can offer information about when, where, and by whom an artwork was made. This is particularly useful in situations that concern an artwork’s authenticity. For example, titanium white, containing titanium dioxide (TiO_2), did not become commercially available until 1916 [2]. The presence of titanium white leads experts to question the authenticity of the painting because the pigment is not one that would have been available to Renoir. Tin yellow was found in the analysis of an illuminated manuscript. This pigment had been used since 1300 AD, so the manuscript could not be dated earlier than this period [3].

In recent decades, many museums have begun to build vast digital libraries of images of their collections. Image processing has been used in painting storage, analysis, and artist identification for many years [4–6]. Actually, artist or painting identification is a complicated process which involves color as well as pigment analysis, paint sampling, and artist’s handwriting in the brushwork.

The purpose of this research is to demonstrate the usefulness of digital color processing for the analysis of oil paint pigments and for the possible uses of indirect dating, authentication, and identification. Image processing technologies and algorithms are being used to analyze a series of pigments representing a traditional and modern palette including pure and mixture pigments in single and multilayer samples.

2 Material and Methods

2.1 Digital Camera

The picture color from cameras could be a little different than the objects’ colors when a camera is used to take the picture. According to the experiments of Image Engineering, FUJIFILM X-T3 has been ranked in the top 2 for color reproduction [7]. That is, the picture color from cameras could be basically the same as objects’ colors using FUJIFILM X-T3.

2.2 Sample Preparation

The division between the traditional palette prior to the nineteenth century and the modern palette post-nineteenth century serves as inspiration for the samples chosen in this experiment. The traditional paints included lead white, ivory (bone) black, Naples yellow, vermilion, ultramarine blue; the modern paints included titanium white, Mars black, cadmium yellow, alizarin crimson, and cobalt blue. The samples were prepared using linseed oil paints provided by RGH Artists’ Oil

Paints, a company known for making quality paint for artists' purposes without using unnecessary extenders [8]. All samples were prepared using 1.33" × 1.33" Blick canvas panels at 1/8" thickness. Paint was applied using a size eight bright brush and a size eight round brush, cleaning with turpentine between uses. Five samples were prepared using flake white as a primer and using pairs of equivalent colors. Each pair of colors was painted individually at the ends of the panel and gradually mixed at the center. This served the purpose of providing spectra of individual pigment composition as well as spectra of the overlap of similar colors with different compositions.

The binary mixtures consisted of same colors from traditional and modern paint (i.e., ivory white with titanium white, etc.); this produced five different samples of the same color. The others were mixtures that produced different colors; they consisted of (1) alizarin red with cadmium yellow; (2) cadmium yellow with lead white; (3) cobalt blue with titanium white; (4) titanium white with ivory black; (5) Naples yellow with ultramarine blue; and (6) vermilion with cobalt blue.

2.3 Image Histogram

A color histogram is a representation of the distribution of colors in an image, which represents the number of pixels that have colors in each of a fixed list of color ranges. The color histogram in this research is based on RGB color space.

There are several algorithms to convert RGB color to grayscale. We use $\text{grayscale} = (R + G + B)/3$ formula in this research. An image histogram is a representation of the tonal distribution. For the grayscale histogram of an image, the horizontal axis of the graph represents the tonal variations, while the vertical axis represents the number of pixels for each tonal value.

The bins (0–255) are plotted on the X-axis. And the Y-axis counts the number of pixels in each bin. 0 is corresponding to black; 255 is white. Any color can be expressed as the distribution among 0–255.

The left side of the horizontal axis represents the black and dark areas, the middle represents medium gray, and the right-hand side represents light and pure white areas.

3 Experiment Results

FUJIFILM X-T3 has been used to take pictures of all pigments after painting on canvas. Figure 1 shows the picture of mixture pigment (alizarin red with cadmium yellow) and pure pigment (alizarin red) as well as pure pigment (cadmium yellow). Figure 2 shows the picture of mixture pigment (vermilion with cobalt blue) and pure pigment (cobalt blue) as well as pure pigment (vermilion).

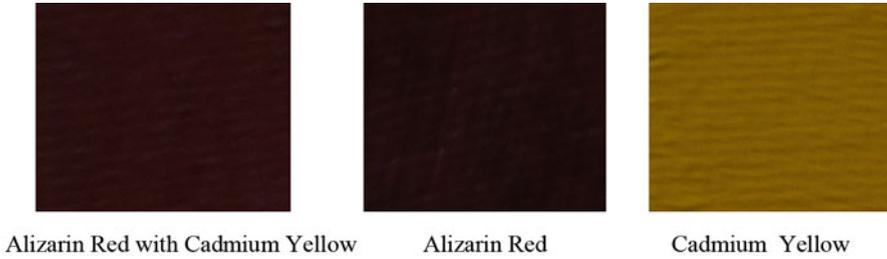


Fig. 1 Shown is the picture of mixture pigment (alizarin red with cadmium yellow) and pure pigment (alizarin red) as well as pure pigment (cadmium yellow)



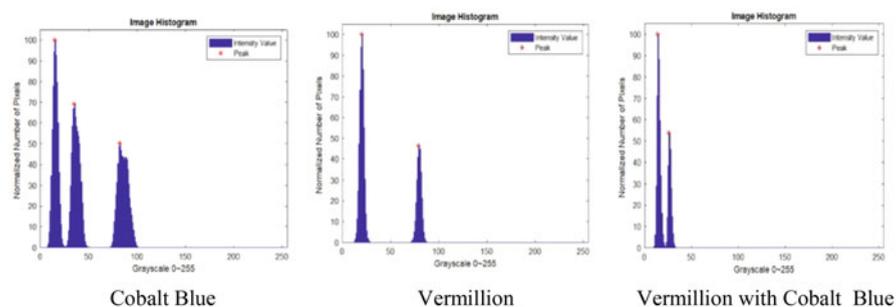
Fig. 2 Shown is the picture of mixture pigment (vermillion with cobalt blue) and pure pigment (cobalt blue) as well as pure pigment (vermillion)

Table 1 lists the peak locations of 21 pigment images' histograms. Table 1 shows that the peak location of a pure pigment histogram is totally different from the peak location of a mixture pigment (containing two pure pigments) histogram. For instance, the peaks of pure pigment (cobalt blue) locate scale values 14, 34, and 81; the peaks of pure pigment (vermillion) locate scale values 19 and 78; the peaks of mixture pigment (vermillion with cobalt blue) histogram locate gray scale values 13 and 25. Also, Fig. 3 displays the histogram of mixture pigment (vermillion with cobalt blue) and of pure pigment (vermillion) as well as pure pigment (cobalt blue).

According to the above results, there are three peaks in the histogram of pure pigment—cobalt blue; two peaks in the histogram of pure pigment—vermillion, and two peaks in the histogram of mixture pigment (vermillion with cobalt blue). So, we can easily tell these pure pigments and mixture pigments apart since the number of peaks and the peak locations of histograms of pure and mixture pigments are different. For instance, two peak locations (19 and 78) of a histogram of pure pigment (vermillion) have a wide interval between two peaks. However, the two peak locations of the histogram of mixture pigment (vermillion with cobalt blue) are very close to each other. Thus, pure pigment and mixture pigment can be easily separated via their image histograms.

Table 1 The peak locations of 21 pigments' image histograms

Name of pigment	Peak location in histogram
Alizarin red with cadmium yellow	15 42
Alizarin red	12 30
Alizarin red with vermilion	23 31 64
Cadmium with Naples yellow	4 54 86
Cadmium yellow with lead white	81 107
Cadmium yellow	91 126
Cobalt blue	14 34 81
Cobalt blue with titanium white	82 163
Ivory black	14
Ivory with mars black	6 13 19
Lead white	138
Lead with titanium white	118 151 177
Mars black	12 16
Naples yellow	21 61 87
Naples yellow with ultramarine blue	24 31
Titanium white	132 146
Titanium white with ivory black	30 48 64
Ultramarine blue	6 13 49 60
Ultramarine with cobalt blue	6 19 72
Vermillion with cobalt blue	13 25
Vermillion	19 78

**Fig. 3** Shown is the image histogram of mixture pigment (vermilion with cobalt blue) and histogram of pure pigment (vermilion) as well as pure pigment (cobalt blue)

4 Conclusions

In this paper, we have investigated the peak locations of image histograms of 21 fundamental pigments containing pure pigments and mixture pigments. From the experimental results and analyses, we have concluded that, whether pure pigments or mixture pigments, the pigments' histograms have their unique peak locations. Our research indicates that fundamental pigments can be effectively distinguished and separated according to the peak locations of their pigments' image histograms.

References

1. P. Hunt, "Revolution in paint." North Carolina Museum of Art, 2006, pp. 1–16
2. Titanium dioxide white, pigments through the ages, <http://www.webexhibits.org/pigments/indiv/history/tiwhite.html>. Date accessed 7 Aug 2017
3. K. Melessanaki, V. Papdakis, C. Balas, D. Anglos, Laser induced breakdown spectroscopy and hyper-spectral imaging analysis of pigments on an illuminated manuscript. *Spectrochim. Acta B* **56**, 2337–2346 (2001)
4. C.R. Johnson Jr., E. Hendriks, I.J. Berezhnoy, E. Brevdo, S.M. Hughes, I. Daubechies, J. Li, E. Postma, J.Z. Wang, "Image processing for artist identification", July 2008, *IEEE signal processing magazine*, pp. 37–48
5. M. Barni, A. Pelagotti, A. Piva, Image processing for the analysis and conservation of paintings: Opportunities and challenges. *IEEE Signal Processing Mag* **22**(5), 141–144 (2005)
6. H. Maitre, F. Schmitt, C. Lahanier, 15 years of image processing and the fine arts. *Proc. IEEE Int. Conf. Image Processing* **1**, 557–561 (2001)
7. <https://www.pdnonline.com/gear/cameras/the-best-cameras-for-color-reproduction-ranked/>
8. RGH artists' oil paints, website: <http://www.rghartistoilpaints.com/>

Part V
**High-Performance Computing, Parallel
and Distributed Processing**

Toward a Numerically Robust and Efficient Implicit Integration Scheme for Parallel Power Grid Dynamic Simulation Development in GridPACK™



Shuangshuang Jin, Shirang G Abhyankar, Bruce J Palmer, Renke Huang, William A Perkins, and Youso Chen

1 Introduction

The need for accelerating power grid simulation through high-performance computing (HPC) for the future power grid has long been recognized. While nowadays most efforts are focused on constructing a particular HPC-based standalone parallel application, GridPACK™ (Grid Parallel Advanced Computational Kernels) has emerged as a unique framework to provide advanced open-source parallel computing kernels to support multi-application development across a broad range of power system models and algorithms [1]. This is due to its tailored design in making extensive use of software templates and generic modules to provide high-level functionality and software reusability [2].

Dynamic simulation is a central application in GridPACK to perform transient stability analysis and dynamic security assessment in power systems. Speeding up dynamic simulation, through fast and scalable solution of the power grid differential-algebraic equations (DAEs), is not only to transient stability assessment itself but also to a series of subsequent applications that stem from dynamic simulation such as dynamic contingency analysis, real-time path rating, etc.

The existing dynamic simulation module developed in GridPACK uses a customized second-order modified Euler (ME) explicit numerical integration scheme to integrate the system dynamics of generating units, controllers, and other dynamic

S. Jin (✉)

Clemson University, North Charleston, SC, USA

e-mail: jin6@clemson.edu

S. G. Abhyankar · B. J. Palmer · R. Huang · W. A. Perkins · Y. Chen

Pacific Northwest National Laboratory, Richland, WA, USA

e-mail: bruce.palmer@pnnl.gov; renke.huang@pnnl.gov; william.perkins@pnnl.gov; yousu.chen@pnnl.gov

devices, e.g., generators, exciters, governors, loads, and relays at each time step. The ME integration scheme is a popular choice in industry-grade software due to its simplicity and, mainly, as a remnant of legacy implementations that are too unwieldy to replace. While capable of running large-scale dynamic simulation with detailed models in near real time [3], its model-solver embedded design makes it difficult to separate models from the solver in GridPACK to add new models for complex systems or implement different integration solvers for potentially better computational gains. Therefore, whenever a new device model is needed to represent a new system, not only the first-order differential equations describing the dynamics of this device model need to be derived, but also each phase of the alternating explicit solution approach (e.g., the predication and correction of current injections and device state variables) has to be explicitly implemented within the proximity of this device model to ensure the model consistency and integrity. On the other side, whenever a new integration method is deployed for further speedup, there is no quick turnaround other than rewriting the whole deck of device models to reflect the new solution scheme, which has so far become a major hurdle to leveraging other numerically robust and efficient solvers in GridPACK.

This paper presents an ongoing research effort toward incorporating a numerically robust and efficient “variable step” implicit integration scheme to promisingly achieve high computational performance in power system dynamic simulation. With the design of a dedicated DAE model/solver interface leveraging the PETSc library [4] and a subsequent restructure of device classes, the newly developed dynamic simulation module is able to (1) separate models and solvers so that changes made to one do not affect the other and (2) provide the flexibility to select a variety of integration methods, including implicit time-stepping solvers, through configuration options. Therefore, there is a promise of improved computational performance and usability (e.g., by taking larger time-steps to dramatically reduce the simulation time utilizing the time-step adaptivity of implicit time-stepping solvers) without sacrificing the numerical accuracy.

The remainder of this paper is organized as follows: Sect. 2 provides a brief overview of the GridPACK software framework and its existing ME integration-based dynamic simulation implementation. Section 3 introduces the new DAE solver interface for dynamic simulation to enable generalized numerical integration especially implicit time-stepping, followed by an illustration of the subsequent design on new device base and model classes in GridPACK, as compared to the ME implementation, so as to leverage the new DAE solver interface. Section 4 provides the details of a working example with preliminary testing results as a proof-of-concept demonstration of the feasibility and validity of the new DAE model/solver design. Section 5 concludes the paper with future work toward delivering a fully functional advanced DAE scheme capable of running large-scale dynamic simulations in real time.

2 GridPACK Software Framework

GridPACK™ is a C++-based object-oriented software framework for developing power grid simulations on high-performance computing platforms [1]. It contains modules for creating distributed representations of power grid networks, supporting distributed matrices and vectors, supplying linear and nonlinear solvers, and mapping data between networks and matrices [5]. Objects in GridPACK context are defined as components (e.g., buses, branches, and generators, etc.) in a power grid to naturally provide and maintain input data to power grid applications.

The major modules in GridPACK include the following:

- (1) Network module, which allows users to include arbitrary models for the power grid bus and branch components in the network to generate appropriate equations for power grid applications. The network module supports distribution of the power grid network over multiple processors through a partitioner, manages data exchanges between ghost buses and ghost branches (which are copies of buses and branches that are located on other processors but are connected to locally owned buses) across partitioned networks, and provides access to buses and branches on each processor.
- (2) Mapper module, which generates distributed matrices from partitioned network and determines the dimensions of the matrix and the locations of all matrix elements across processors after user supplies the matrix elements contributed by each network component.
- (3) Math module, which supports the creation and manipulation of distributed matrices and vectors and supplies the linear and nonlinear solvers used by applications to implement solution algorithms.
- (4) Application module, which is designed to be called as parts of other programs and allow developers to string several different types of calculations together to simplify the task of creating more complicated workflows in which the results of one type of calculation are fed into the input of another.

More details of the overall design of GridPACK framework and its modules are described in [2].

Dynamic simulation, due to its importance to power grid transient stability analysis and close correlation with other applications, is converted to a module in GridPACK. The module integrates the equations of motion using an algorithm based on the inversion of the “full Y-matrix” [5]. Mathematical models in terms of DAEs as shown in (1)

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{0} = \mathbf{g}(\mathbf{x}, \mathbf{u}) \end{cases} \quad (1)$$

are developed to describe the dynamic behavior of various system components, where \mathbf{x} represents dynamic state variables of generators and their controllers and \mathbf{u} represents algebraic variables for the coupled network between generators, loads, and the transmission system. This model has been designed to enable the addition of generator models (e.g., *GENSAL*, *GENROU*) that extend beyond the classical generator (*GENCLS*) as well as several other devices including exciters (*EXDCI*, *ESSTIA*), turbine governors (*WSIEGI*, *WSHYGP*), relays (*LVSHBL*, *FRQTPAT*, *DISTR1*), and dynamic loads (*ACMTBLU1*, *IEEL*, *MOTORW*, *CIM6BL*) [6].

Each of these devices has a base class (interface) and a model class (implementation). The base classes specify certain functions that must be implemented by the model so that it can interact with other GridPACK modules, while the model class implements functionality outside the base class that is unique to the particular model.

2.1 Existing Design of Device Base Class

The existing dynamic simulation module implements the “full Y-matrix” algorithm using the ME integration scheme. In this design, a device base class is implemented as a self-contained class independent of any other base module available in GridPACK. It implements all the methods required to perform the alternating explicit ME integration (e.g., *INorton*, *NortonImpedence*, *predictor_currentInjection*, *correction_currentInjection*, *prediction*, *correction*, etc.) rigidly inside the model as shown below, before it pushes out all the updated device state variables to the system-level control flow:

```
class BaseDeviceModel{
public :
    void load(const boost :: shared_ptr < gridpack ::
        component :: DataCollection > data, int idx);
    void init(gridpack :: ComplexType * values);
    gridpack :: ComplexType INorton();
    gridpack :: ComplexType NortonImpedence();
    void predictor_currentInjection(bool flag);
    void corrector_currentInjection(bool flag);
    void predictor(double t_int, bool flag);
    void corrector(double t_inc, bool flag);
```

```

    void setVoltage(gridpack :: ComplexType voltage);
    ... other class methods
protected :
    ... class variables
}

```

Although a key advantage of this solution approach is that the nonlinear solution is only done for the network algebraic equations (since the dynamic variables do not need to be solved but explicitly updated as compared to implicit integration solutions), it does not provide enough flexibility to switch out the ME solver and replace it with another Numerically robust and efficient solver due to its rigid structure. Significant coding efforts have to be made explicitly in each GridPACK dynamic simulation device to make it adaptable.

2.2 Existing Design of Device Model Class

In the existing scheme, a device model class implements the update for each state variable $x_{n+1} = x_n + \frac{\Delta t}{2}(f_n + f_{n+1}(\tilde{x}_{n+1}))$ explicitly inside its prediction and correction steps, where $\tilde{x}_{n+1} = x_n + \Delta t f_n$ is the approximation or prediction for \tilde{x}_{n+1} . A standard API is provided so that the update equations for each device model's own differential variables can be implemented. Again, this "fixed-step" predictor-corrector alteration-based implementation has the model classes tightly coupled with the solver, therefore further reducing the flexibility to update solvers. To enable separation of the model and the solver in GridPACK dynamic simulation, changes to both the device base and model classes are required.

3 New DAE Solver Interface

The DAE solvers are designed to solve systems of the form $F(\mathbf{X}, \dot{\mathbf{X}}) = 0$, where \mathbf{X} is a vector comprising both dynamic variables \mathbf{x} and algebraic variables \mathbf{u} . Unlike the existing ME implementation which contains its intermediate calculations inside the device classes, the DAE solvers keep track of all intermediate states in the calculation. To use the solvers, the GridPACK implementation must supply functions that map the state vector \mathbf{X} to an internal state of the network and functions that calculate, for a given network state \mathbf{X} , the time derivatives $\dot{\mathbf{X}}$ and Jacobians corresponding to that state. Once these functions are supplied, a wide variety of solvers can be easily accessed through DAE solver interface. Objects of type *JacobianBuilder* and *FunctionBuilder* representing these two functions are passed to the constructor of the nonlinear solver.

PETSc's DAE solvers are integrated in GridPACK for applications to use via an object `gridpack::math::DAESolver` as below:

```
daesolver = new gridpack :: math :: DAESolver(comm,
lsize, daejacobian, daefunction)
```

where *comm* is the communicator, *lsize* is the local size of the solution vector, and *daejacobian* and *daefunction* are the Jacobian and residual function objects being passed to the constructor of the nonlinear solver:

```
gridpack :: math :: DAESolver :: JacobianBuilder
daejacobian = boost :: ref(*this);
gridpack :: math :: DAESolver :: FunctionBuilder
daefunction = boost :: ref(*this);
```

Here, the *daejacobian* and *daefunction* are pointing to two overloaded functions as declared in the following form:

```
void operator()(const double& time,
const gridpack :: math :: Vector& X,
const gridpack :: math :: Vector& Xdot,
gridpack :: math :: Vector& F);
void operator()(const double& time,
const gridpack :: math :: Vector& X,
const gridpack :: math :: Vector& Xdot,
const double& shift, gridpack :: math :: Matrix& J);
```

where *X* is the solution vector representing the current system state, *Xdot* is the vector of derivatives, *F* is the residual vector, *J* is the Jacobian matrix, and *shift* is a scalar provided by PETSc to be used in the Jacobian calculation. The benefit of using the references command instead of making a copy is that it helps preserve any state that might be present in the overload function between invocations of the functions *daejacobian* and *daefunction* by the solver.

The Jacobian and residual functions are called by PETSc at every time step, each of which comprises three major parts:

1. Push current values in *X* and *Xdot* from the overall system-level vectors back into the network components (buses and the device).
2. Update the ghost buses so that nonlocal bus variables get updated.

3. Evaluate the DAE Jacobian and residual (compute the bus, branch, and device contributions for the residual and Jacobian function), and set them in the system-level vector F and matrix J .

Once the *daesovler* object is created and set up, the methods that pertain to the *DAESolver* can be called from the dynamic simulation module's main control flow across all fault periods (pre-fault, on-fault, and post-fault) with high consistency and simplicity. For example, an *initialize()* method to initialize system state variables and a *solve()* to run the time-stepping dynamic simulation till *endtime* are shown below:

```
gridpack :: math :: DAESolver :: initialize(double
    starttime, double timestep, gridpack :: math
    :: Vector& X);
gridpack :: math :: DAESolver :: solve(double endtime,
    double maxsteps);
```

3.1 New Design of Device Base Class

A major redesign in the new dynamic simulation module is to have the device base class inherit GridPACK's *BaseComponent* class *gridpack::component::BaseComponent* from the network module as given below:

```
class BaseDeviceModel : public gridpack ::
    component :: BaseComponent {
public :
    void setMode(int mode){p_mode = mode; }
    void load(const boost :: shared_ptr < gridpack ::
        component :: DataCollection > data, int idx);
    void init(gridpack :: ComplexType * values);
    bool vectorSize(int * isize) const;
    void setValues(gridpack :: ComplexType * values);
    bool vectorValues(gridpack :: ComplexType * values);
    bool matrixDiagEntries(int * nval, int * row, int
```

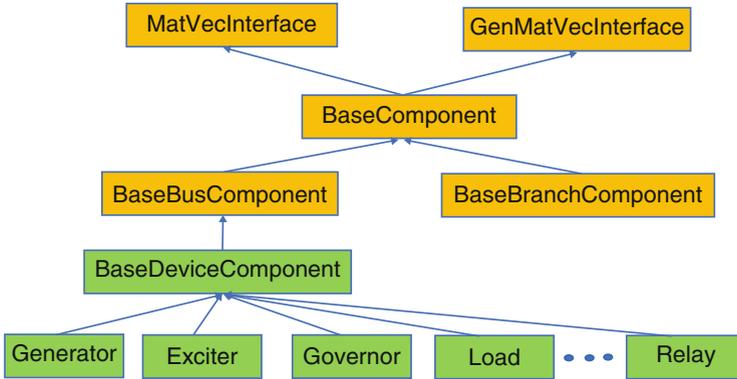


Fig. 1 Schematic diagram showing the interface hierarchy with newly added network components (in green)

```

    * col, gridpack :: ComplexType * values);
    ... other class methods
protected :
    ... class variables
}

```

As shown in Fig. 1, each device is added to the GridPACK framework as a new network component. There are two main advantages to this revision: First, GridPACK's *BaseComponent* class provides virtual methods (e.g., *load*, *setValues*, *vectorValues*, *matrixDiagEntries*, etc.) for loading data from *DataCollection* objects and setting contributions from components into the vector/matrix, and vice-versa. Inheriting from the *BaseComponent* allows the device model to inherit these methods and return its contributions, which are needed for the residual vector and Jacobian matrix calculation. Second, GridPACK has mapper objects *BusVectorMapper* and *FullMatrixMapper* that map the contributions from buses/branches to vectors/matrices. Inheriting from the *BaseComponent* allows usage of the mapping from the mapper objects in mapper module. The *MatVecInterface* class can be further utilized to eliminate a large number of tedious and error-prone index calculations that would otherwise need to be performed in order to determine where in a matrix a particular data element should be placed.

For further clarification, an example of the implementation of the *setValues* method in the application *Bus* class for the residual evaluation is shown below:

```

void DsimBus :: setValues(gridpack :: ComplexType * value{
    ...
    * p_VDQptr = real(values[0]);
    * (p_VDQptr + 1) = real(values[1]);
    ...
    genvals = values + 2;
    for (i = 0; i < p_nngen; i ++){
        p_gen[i] → setValues(genvals);
        if (has_ex) p_gen[i] → getExciter() → setValues(genvals);
        if (has_gv) p_gen[i] → getGovernor() → setValues(genvals);
    }
}

```

In this method, the bus sets the internal values of the voltage magnitude and phase angle pointed by *p_VDQptr* first and then invokes the *setValues* method of generators, exciters, and governors subsequently if there exists any of these devices on this bus to collect all their contributions to the residual evaluation vector. Upon completing this call, the overall residual vector at the system level as a combination of bus voltages and dynamics of all its devices will be available for the DAE solver. To obtain the total number of variables at the system level, a new method *vectorSize* in each model class is added to return the number of state variables for that model.

The offset from where each device inserts its contribution to the overall residual vector is controlled via a pointer by retrieving and adding up the *vectorSize* of each coupled device. The processing of all bus/device vector blocks is executed in parallel in GridPACK. A conceptual configuration of the residual vector contribution from one bus coupled with a *GENROU* generator, an *ESSTIA* exciter, and a *WSIEGI* turbine governor is listed in Table 1. The contributions of other buses with/without devices will be processed in the same manner simultaneously and get appended to the end of its preceding bus's vector block based on their bus indices.

3.2 New Design of Device Model Class

With the new design, each individual device model class inherits its base class and only implements its own device equations to provide their specific contribution to the system-level residual vector and Jacobian matrix from this device. As an example, the *GENCLS* generator model and the *GENROU* generator model both inherit from *BaseGenModel*, but their dynamics are each formulated individually

Table 1 A conceptual configuration of residual vector contribution from one bus

Vector index	Device		
	Type	Name	Variables
0	BUS	Voltage magnitude	VD
1	BUS	Phase angle	VQ
2	GENROU	Rotor angle	x1d
3	GENROU	Rotor speed	x2w
4	GENROU	Transient Q axis Eq	x3Eqp
5	GENROU	Transient D axis flux	x4Psidp
6	GENROU	Transient Q axis flux	x5Psiqp
7	GENROU	Transient D axis Ed	x6Edp
8	ESSTIA	Va	x1Va
9	ESSTIA	Sensed voltage	x2Vcomp
10	ESSTIA	Lead lag State 1	x3LL1
11	ESSTIA	Lead lag State 2	x4LL2
12	ESSTIA	Derivative feedback	x5Deriv
13	WSIEG1	Vm	x1LL
14	WSIEG1	Sensed voltage	x2GovOut
15	WSIEG1	Turbine 1	x3Turb1
16	WSIEG1	Turbine 2	x4Turb2
17	WSIEG1	Turbine 3	x5Turb3
18	WSIEG1	Turbine 4	x6Turb4

in their *vectorValue* method. For example, the residual evaluation for GENCLS generator model is shown below:

$$\begin{aligned}
 values[\delta_idx] &= p_dw / OMEGA_S - p_deltadot; \\
 values[dw_idx] &= (p_Pm - VD * p_Ep * \sin p_delta / p_Xdp \\
 &\quad + VQ * p_Ep * \cos p_delta / p_Xdp \\
 &\quad - p_D * p_dw) / (2 * p_H) - p_dwdot;
 \end{aligned}$$

where *delta*, *dw*, *deltadot*, and *dwdot* are the generator state variables and their derivatives of the GENCLS generator model, *VD* and *VQ*, are the voltage variables, *_idx* denotes the index of a specific state variable, and *p_* denotes the pointer to a variable. Others are system constants, generator parameters, or intermediate variables.

The Jacobian matrix is derived based on the given device equations as shown above and implemented in the *matrixDiagEntries* method inside the device model class, if an analytical Jacobian (AJ) matrix is to be used instead of a finite difference Jacobian approximation (FDJA) matrix that is automatically supplied by PETSc by default. The *matrixDiagEntries* method returns the Jacobian matrix entries of each device model. Taking the GENCLS model as an example, let $\mathbf{x} = [\delta, dw]^T$ denote the generator state variable vector and $\mathbf{y} = [VD, VQ]^T$ denotes the current

injection variable vector. Suppose $\mathbf{u} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$, $\mathbf{G} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}$, where \mathbf{f} denotes the equation for the residual evaluation function and \mathbf{g} denotes the equation of generator current injection function as given below analytically:

$$IGD+ = (-VQ + p_{Ep} * \sin(p_delta))/p_Xd p;$$

$$IGQ+ = (VD - p_{Ep} * \cos(p_delta))/p_Xd p;$$

IGD and IGQ are the real part and imaginary part of the generator current. The Jacobian matrix of the DAE equations can be derived as

$$\mathbf{J}(\mathbf{G}) = \mathit{shift} \frac{\partial \mathbf{G}}{\partial \dot{\boldsymbol{\mu}}} + \frac{\partial \mathbf{G}}{\partial \boldsymbol{\mu}} = \begin{bmatrix} \mathbf{J}_{fx} & \mathbf{J}_{fy} \\ \mathbf{J}_{gx} & \mathbf{J}_{gy} \end{bmatrix}$$

where \mathbf{J}_{fx} and \mathbf{J}_{fy} represent the partial derivatives of generator equations with regard to generator variables and voltage variables and \mathbf{J}_{gx} and \mathbf{J}_{gy} represent the partial derivatives of the generator currents with regard to the generator variables and voltage variables, respectively. The explicit equations for these submatrices can be derived from the equations of motion of the dynamical variables as shown below:

$$\mathbf{J}_{fx} = \begin{bmatrix} -\mathit{shift} & \frac{1}{\text{OMEGA} \cdot S} \\ 0 & \frac{(-VD * p_{Ep} * \cos p_delta - VQ * p_{Ep} * \sin p_delta)}{p_Xd p} \frac{1}{2 * p_H} \end{bmatrix}$$

$$\mathbf{J}_{fy} = \begin{bmatrix} 0 & 0 \\ \frac{-(p_{Ep} * \sin p_delta)}{p_Xd p} \frac{1}{2 * p_H} & \frac{(p_{Ep} * \cos p_delta)}{p_Xd p} \frac{1}{2 * p_H} \end{bmatrix}$$

$$\mathbf{J}_{gx} = \begin{bmatrix} \frac{p_{Ep} * \sin p_delta}{p_Xd p} & 0 \\ \frac{p_{Ep} * \cos p_delta}{p_Xd p} & 0 \end{bmatrix}$$

$$\mathbf{J}_{gy} = \begin{bmatrix} \frac{1}{p_Xd p} & 0 \\ 0 & \frac{-1}{p_Xd p} \end{bmatrix}$$

The details of the residual evaluation and analytically calculated Jacobian matrices of the *GENROU* model are omitted here due to the lengthy formulation and the page limit.

4 Validation and Preliminary Results

A small working example is built to show the validity of the new integration scheme as a proof-of-concept demonstration. This model represents a simplified structure of a 9b3g system, consisting of 9 buses and 3 synchronous generators [7].

The generator model can be either a simplified *GENCLS* model or a *GENROU* generator model that is capable of interacting with an *ESSTIA* exciter model (IEEE (1992/2005)-type ST1A excitatory model) and/or a *WSIEG1* turbine-governor model (IEEE steam turbine/governor model with deadband and nonlinear valve gain added) to mimic a more complex system. The simulation length is 30 s with a time step of 5 or 50 ms, depending on the test. A bus fault at bus 1 is introduced at 1.0 s and cleared at 1.05 s.

4.1 Accuracy

The accuracy of the GridPACK dynamic simulation module is compared against a widely used sequential computing-based commercial software tool PowerWorld Simulator [6] with *GENROU* generator model for the 9g3b system. Figure 2 shows how each bus with generator device distributively contributes to the formulation of the analytical Jacobian matrix. The element differences between the explicitly derived AJ and the automatically supplied FDJA are all within 10^{-7} .

Figure 3 shows the simulation outputs on the faulted bus of the system. The curve of pre-fault and on-fault periods shows perfect matches between our GridPACK simulation and the commercial PowerWorld Simulator, indicating the network

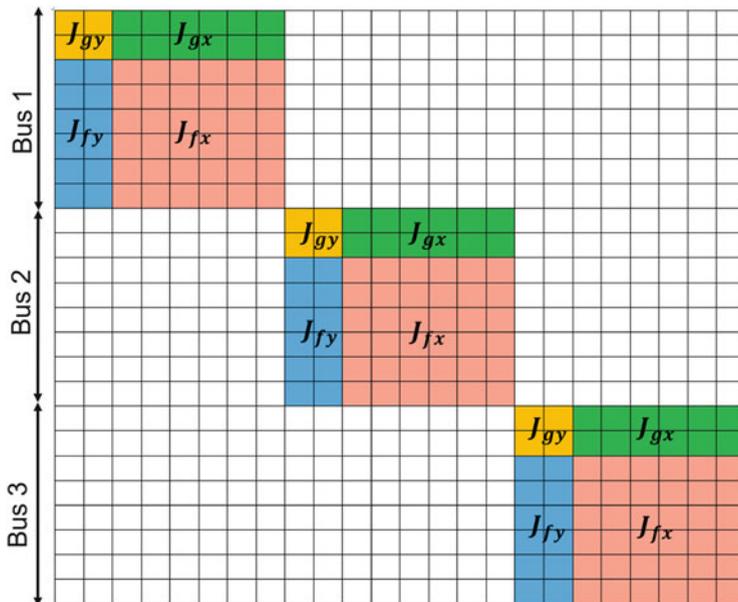


Fig. 2 The Jacobian matrix component contribution from the three generator buses in the 9b3g system (Non-generator buses are omitted here)

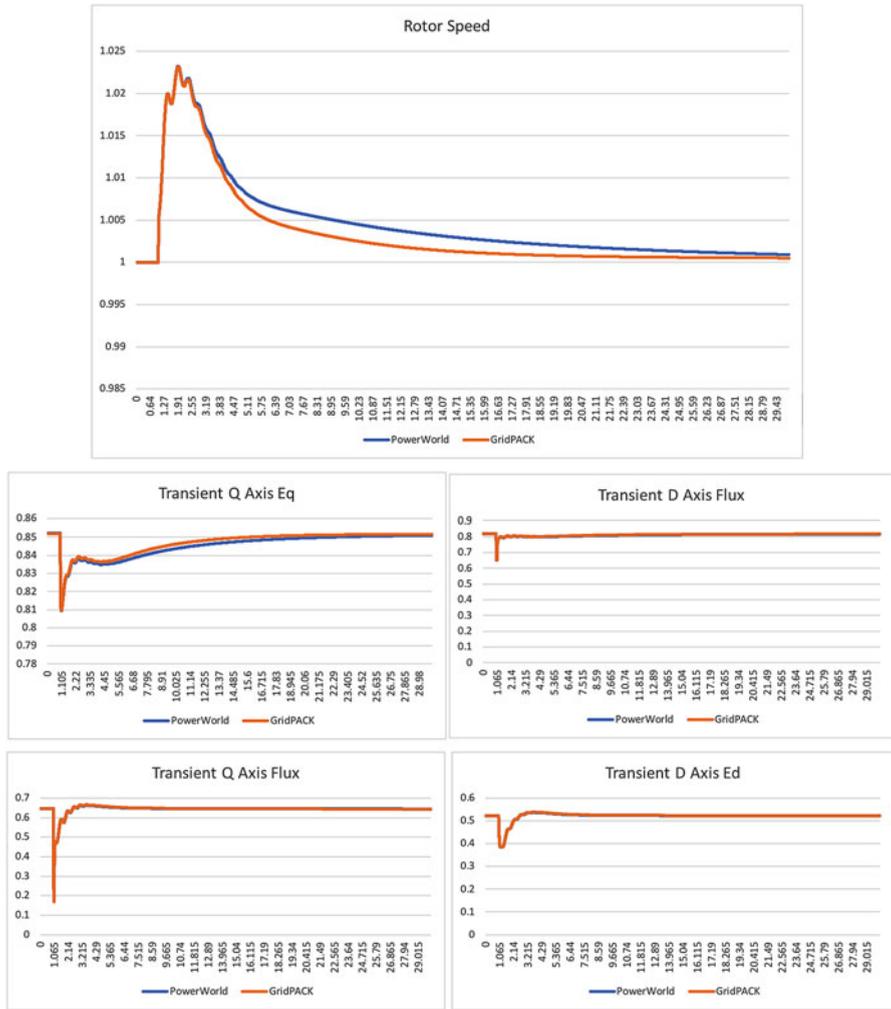


Fig. 3 Comparison of generator dynamics on a faulted bus of the 9b3g system *GENROU* model against PowerWorld Simulator

interface calculation of generator models and network solutions are accurate. The slight differences in post-fault period may come from the different numerical integration methods (PowerWorld uses forward Euler and second order Runge-Kutta methods) and the precision of the codes (GridPACK uses double precision, while PowerWorld uses single precision).

4.2 Flexibility

The ability to evaluate different integration approaches for the dynamic simulation module in GridPACK, with a special focus on adaptive “variable step” time-stepping solvers from PETSc, is demonstrated by specifying different configuration options in an XML-formatted input file that supports simple hierarchical data format. This format is compatible with many other types of software tools and also provides good flexibility and extensibility. The choice of solvers can be specified in the *NonLinearSolver* option block and run without changing any of the underlying source code or input data format, which enables users to experiment with a variety of solvers. Shown below is a segment of setup of a Scalable Nonlinear Equations Solvers (SNES) using FDJA through PETSc Options in the XML configuration for a typical dynamic simulation run in GridPACK:

```

< NonlinearSolver >
  < SolutionTolerance > 1.0E - 05 < /SolutionTolerance >
  < MaxIterations > 50 < /MaxIterations >
  < PETScOptions >
    - dsim_snes_fd
    - dsim_snes_monitor
    - dsim_ts_monitor
  < /PETScOptions >
< /NonlinearSolver >

```

4.3 Performance

The preliminary study on the performance of the new implementation falls into two parts:

1. Benchmark a number of PETSc’s time-stepping solvers on *GENCLS* model to evaluate the speedup with reduced time steps. Table 2 shows the results of running this study with a list of implicit integration methods. Differences in execution time for “variable step” algorithms are compared to a “fixed-step” algorithm. Large steps are taken during portions of the simulation and reduce the total number of steps for integration. The execution time is reduced correspondingly.
2. Test the speedup of solving the DAEs using AJ matrices versus FDJA. Table 3 shows the results of running a series of tests with the 9g3b system having the *GENROU* generator model. The AJ-based implicit time-stepping consistently

Table 2 Benchmark of PETSc's time-stepping solvers on *GENCLS* model-based dynamic simulation in GridPACK

Method	Order	Variable step	Nsteps	Execution time (s)
Trapezoidal	2	No	1000	2.43
Trapezoidal	2	Yes	131	0.3775
Implicit RK	2	Yes	130	0.637
Implicit RK	3	Yes	130	0.79
Implicit RK	4	Yes	130	1.37
Rosenbrock	2	Yes	130	0.45
Rosenbrock	3	Yes	130	0.72

Table 3 Time statistics of running a 30-s dynamic simulation on 9b3g system with FDJA vs. AJ implementations

Average time spent (s)	FDJA	AJ
Time step = 5 ms	337.5911	201.8997
Time step = 50 ms	34.634	19.9131

shows better speed performance than FDJA regardless of the variation in time steps, which showcases the value of deriving analytical Jacobians for all device models for further speedup as a next step.

This preliminary study is focused on putting together a set of representative models for essential tests in GridPACK to evaluate the newly designed dynamic simulation module's feasibility and extensibility to leverage different time-stepping solvers. Many of the detailed device models and their analytical Jacobians are still under development at this stage. Optimization on the performance of the code has not been extensively tuned due to the limited pool of supported models and the small size of the test system so far. It is expected that the performance improvement could be much more significant when the fully developed package is ready to support large-scale complex system dynamic simulations running on multiple processors and ultimately benefit from the partitioning of the network and parallel computing capability provided in GridPACK.

5 Conclusions and Future Work

In this work, a new integration scheme for dynamic simulation in GridPACK is pursued by restructuring the device models leveraging the network module to interface with PETSc's DAE solvers. Validation and preliminary performance results demonstrate favorable values of the design. This is an ongoing effort toward enabling a numerically-robust and efficient dynamic simulation for future dynamic security assessment application development in GridPACK. The successful implementation of this work will improve the computational performance and

usability of a series of parallel power system applications to speed up power grid modeling and simulation for real-time decision support through HPC facilities.

For future work, the authors intend to continue the research effort with the following planned activities:

1. Extend the functionality to more device models to fit the needs of running large-scale realistic systems with complex detailed models.
2. Derive analytical Jacobian matrices for all device models to enable faster evaluation on parallel architecture than finite difference Jacobian approximations.
3. Fine-tune and test the new scheme to achieve real-time computational performance with realistic power systems (e.g., the US Western Electricity Coordinating Council systems) under the parallel computing environment in GridPACK.

References

1. B. Palmer, GridPACK User's Manual Version 3.2 (2019). Available at https://www.gridpack.org/wiki/index.php/File:GridPACK_User_Manual_3.2.pdf
2. B. Palmer, W. Perkins, Y. Chen, S. Jin, D. Callahan, K. Glass, R. Diao, M. Rice, S. Elbert, M. Vallem, Z. Huang, GridPACKTM: a framework for developing power grid simulations on high-performance computing platforms. *Int. J. High Perform. Comput. Appl.* **30**(2), 223–240 (2016)
3. R. Huang, S. Jin, Y. Chen, R. Diao, B. Palmer, Q. Huang, Z. Huang, Faster than real-time dynamic simulation for large-size power system with detailed dynamic models using high-performance computing platform, in *IEEE Power and Energy Society General Meeting (PES)* (2017), pp. 1–5
4. PETSc Users Manual Revision 3.1.2 (2016). Available at <https://www.mcs.anl.gov/petsc/petsc-current/docs/manual.pdf>
5. S. Jin, Y. Chen, R. Diao, Z. Huang, W. Perkins, B. Palmer, Power grid simulation applications developed using the GridPACKTM high performance computing framework. *Electr. Power Syst. Res.* **141**, 22–30 (2016)
6. J. Weber, *Description of Machine Models GENROU, GENSAL, GENTPF and GENTP* (Power-World Corporation, 2015)
7. J. Chow, G. Rogers, Power System Toolbox, Version 3.0

Improving Analysis in SPMD Applications for Performance Prediction



Felipe Tirado, Alvaro Wong, Dolores Rexachs, and Emilio Luque

1 Introduction

During the last few years, high-performance computing (HPC) systems have increased the number of processing units (CPUs) significantly [1]. Nowadays, these systems have an enormous computing power, and everything seems to indicate that this tendency will continue increasing over the coming years, due to constant technological improvements. This makes it possible the increase of the number of cores per processor, as well as the total number of processors in the systems. These improvements have played a significant role in advanced scientific research since they allow us to increase the complexity and the number of experiments.

The gap between maximum performance and performance achieved in scientific applications running in parallel has increased considerably in recent years [2]. The complex architecture of parallel systems, the interdependence between the different components, the communication structures imposed by the algorithm, making concepts such as single program multiple data (SPMD) for dividing tasks and running on multiple processors, which presents a challenge when optimizing performance in distributed memory.

F. Tirado (✉)

Computer Architecture and Operating System Department, Universidad Autónoma de Barcelona, Barcelona, Spain

Departamento de Computación e Industrias, Universidad Católica del Maule, Talca, Chile
e-mail: ftirado@ucm.cl

A. Wong · D. Rexachs · E. Luque

Computer Architecture and Operating System Department, Universidad Autónoma de Barcelona, Barcelona, Spain

e-mail: alvaro.wong@uab.es; dolores.rexachs@uab.es; emilio.luque@uab.es

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_29

387

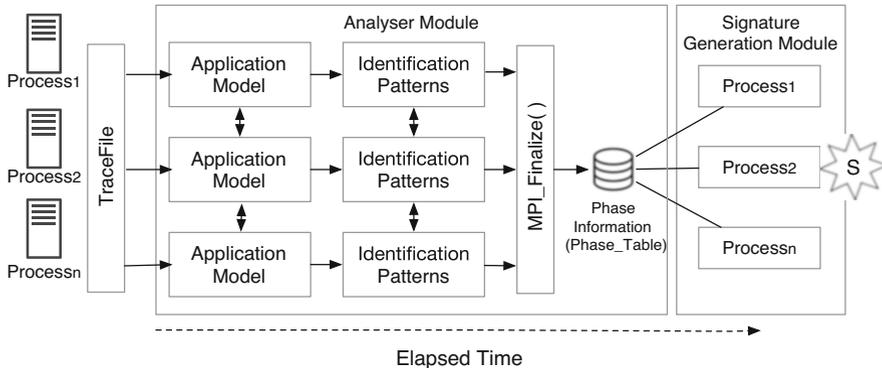


Fig. 1 Parallel analyzer

Tools are available [3–5] that collect and display relevant information about application performance at a high level of abstraction, so that developers can quickly identify and determine the causes that affect application performance.

The PAS2P tool [6, 7] proposes extracting signature performance from MPI applications characterizing the application behavior into phases to predict the execution of the application over the target machine. A phase is a segment of parallel code delimited by MPI communications which are repeated (weight) throughout the execution.

The PAS2P tool consists of two stages: The first stage is the instrumentation and analysis of the application to obtain the phases and the signature generation. The second stage is about the execution of the signature to predict the application execution time (AET).

The PAS2P parallel analyzer module [8], shown in Fig. 1, stores the TraceFile of the application in main memory, generating an independent model of the machine. This is achieved by defining two sections:

1. Application model: In this stage, the events of all the processes are ordered by precedence, taking into account the dependence between them.
2. Identification of patterns: In this stage, the common patterns are identified in a vertical way in all the processes. This allows us to extract relevant information of the performance of each pattern to group them in phases.

The resulting information is stored in a Phase_Table file that will enable us to generate the signature of the application.

The PAS2P parallel analyzer, due to the dependence of its processes, needs the use of collective MPI to synchronize the processes. This affects the performance, when the application scales to a greater amount of processes due to the increase of the number of communications.

The proposed analyzer module makes changes in the way phases are obtained in SPMD applications, as shown in Fig. 2. We can divide it into two sections:

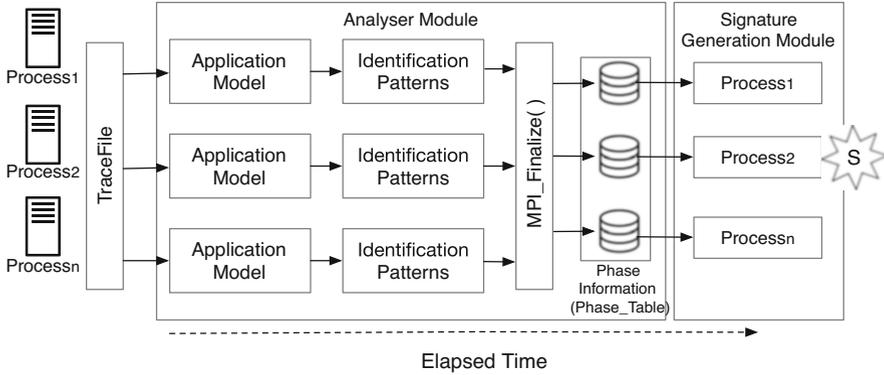


Fig. 2 Proposal parallel analysis approach

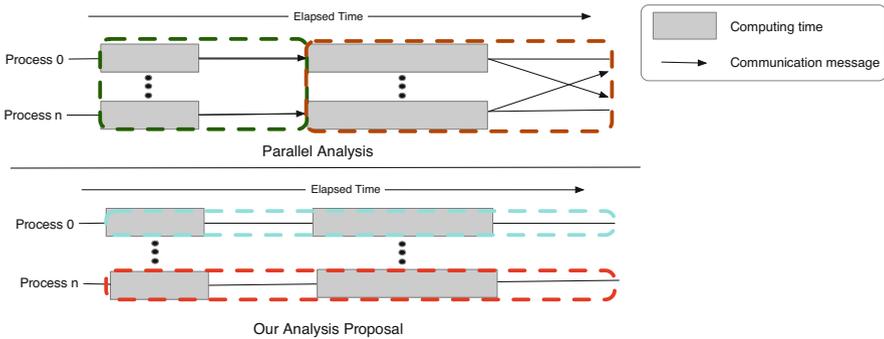


Fig. 3 Types of trace analysis

1. The model of the application: The model is built using logical clocks for each process, permitting us to order the events by precedence, which is achieved by eliminating dependencies between processes.
2. Pattern identification: Each process independently searches for similar behavior patterns to group them into phases and assign them their respective weight. Our proposal eliminates communication between processes.

In order to detect the phases, it is necessary to apply the previous steps in searching for the repetitive segments independently for each process. We then run them as a whole so as to obtain a prediction.

The main difference between the parallel analysis module and our analysis proposal is how the trace of the application is characterized. As shown in Fig. 3, the parallel analysis performs an analysis of the global application, where all processes are interacting, needing communication to synchronize their processes. On the other hand, our proposal makes an independent data analysis between processes, avoiding the use of communication.

To evaluate the quality of our proposal, we performed experiments with different applications, BT and SP of NPB [9] and miniMD [10], increasing the number of processes to verify the behavior of the analyzer and the quality of the prediction. We obtained a measurement of analyzer acceleration up to 11 times greater when compared to the parallel version, thus achieving an average prediction of more than 97%.

In the following section, we present related work. In Sect. 3, we provide an overview of the PAS2P toolkit. Section 4 presents the proposed methodology for SPMD applications. Section 5 deals with experimental results, and in Sect. 6, we present the conclusions and future work.

2 Related Work

There are tools which have also dealt with the scalability limitations that adversely affect the user experience when the tool is used over a large scale. Scalasca [3] is an open-source toolset that can be used to analyze the performance behavior of parallel applications and to identify opportunities for optimization. Although Scalasca can be used over 294 K cores, the version 1.3.0 and underlying versions showed scalability limitations during the collecting and displaying of analysis reports. Developers focused on improving these lacks.

Periscope [5] is a performance analysis tool which analyzes parallel applications to detect performance problems and their causes. This tool overcomes the scalability barrier by performing an automatic distributed online analysis on thousands of processors. The main drawback of Periscope is that the application must be executed entirely to see the optimizations applied.

Paraver [11] uses the CEPBA tool [12] environment to scale the applicability of the Paraver trace visualization and analysis tool to systems with up to several thousand processors. The main approach is based on providing flexible mechanisms to select and summarize the raw data in order to obtain traces that still contain the relevant information with much less data.

MUST [13] is a framework for creating a runtime infrastructure for scalable MPI correctness checking. The main goal is to offer a full set of correctness features for 1000 processes at a runtime overhead of less than 10% and a restricted set of correctness features for 10,000 processes at the same runtime overhead.

The Cray performance analysis tools [14] provide an integrated infrastructure for measurement and analysis of computation, communication, I/O, and memory utilization. It is composed of CrayPat Performance Collector for data capture and Cray Apprentice2 Performance analyzer to a postprocessing data visualization. The Cray performance analysis tools have been used on a large-scale Cray XT system with more than 30,000 processors.

TAU [15] and Vampir [4] have focused their efforts on improving the analysis of data on a large scale. To efficiently achieve this analysis, TAU uses ParaProf parallel performance analyzer [16], which was specifically built for the analysis of large-

scale data. The analysis takes place in memory for fast access and to support global aggregation and analysis views. TAU provides a compressed normalized packed data format, as a container for profile data from any supported measurement tool. This makes the reading of parallel profiles significantly more efficient in ParaProf. Vampir provides efficient access to trace files. This layout allows us to distribute the data in several files, each one storing a “frame” of the execution data. Frames can correspond to a single CPU or to a cluster of CPUs. The frames belonging to a single execution are tied together by means of an index file, thus providing better performance.

On the other hand, A. Morari et al. [17] mention that the next-generation software will have to exploit more programming models to properly use the new generations of HPC systems while maintaining a low overhead. Current trends show that many of the traditional operating system functionalities, such as task planning, load balancing, and interconnection mechanisms, among others, will become increasingly complex to perform at runtime, allowing us to run various applications with different programming models. A typical classification distinguishes between regular and irregular applications. Regular applications have a very regular and predictable pattern of communication and data access, while irregular applications have a very irregular pattern of communication and memory access and, therefore, a very bad spatial and temporal location.

3 PAS2P Overview

The PAS2P tool is based on the repeatability of the parallel application, focusing on the analysis and prediction of MPI application performance using its signature. The PAS2P tool, as shown in Fig. 4, is divided into two stages: the first stage of signature generation and the second stage of signature execution.

The first stage consists of three modules. The first module (instrumentation module) instruments each process of the application by creating one trace file per process. The data files contain the calls to the MPI primitives with their respective hardware counters. The instrumentation of the application MPI is carried out by the dynamic PAS2P library in conjunction with the PAPI library [18], used to measure hardware counters. Finally, PAS2P defines an MPI call as an event associated with the computational data between one MPI call and the next one.

The second module (analyzer module) aims to create a machine-independent application model. To do this, it is necessary to create a global logical clock for all the processes in order to maintain the precedence between the events. The action of receiving or sending an MPI message is defined as an event. The algorithm is inspired by the Lamport [19] method in which if a process sends a message in a logical time (LT), its reception will arrive in a time $LT + 1$.

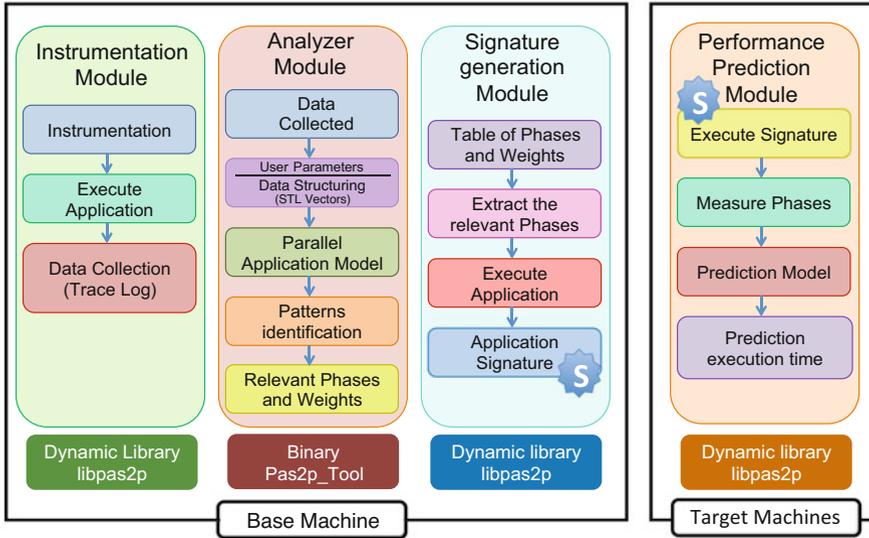


Fig. 4 Stages of the PAS2P methodology

There are two approaches to the analysis method [7, 8], which are detailed below:

- The serial analyzer approach [7] processes data from all application processes into a single collection structure to create a logical global clock and maintain precedence between communication events. When the application runs with a large number of processes, it may result in insufficient memory on the node. Thereby, having to load these data from the swap memory considerably increases the analysis execution time, or in many cases, its execution is not even possible.
- The parallel analyzer approach [8] has been developed using message passing to take advantage of distributed memory. This module allows us to use the PAS2P toolkit on a large scale, achieving an efficient analysis, since it divides the analysis of the data among all the resources and it executes using the same number of resources that the application used for its execution.

Once we have the phases, they are stored in the file called Phase_Table, which will be the output of the analyzer module. To create the signature of the application, each process will read the file Phase_Table, which will contain all the phases and weights of the analyzed application. With this information, it will be possible to predict the execution time of the application on a target machine.

In our proposal, we will model a new approach to the analyzer module, reducing the cost of communication between processes to achieve a reduction in analysis time.

4 Proposed Methodology

PAS2P's parallel analysis module characterizes the application by generating a model of the application independent of the machine, generating a signature that contains information about its phases and weights. This approach presents dependency between the different processes that need to be synchronized, reducing its performance mainly when the application scales. We solved the problem by proposing an extension to the parallel analysis approach, eliminating the data dependencies between processes in SPMD applications, which we call the "Extension of Parallel Analysis to SPMD" (EPAS).

As shown in Fig. 5, we designed a proposal called EPAS that eliminates the dependence of events between processes in the SPMD applications, allowing us to reduce the communication for synchronization reasons. Our proposal is divided into two sections, application model and pattern identification, which are explained in later sections.

When all the processes complete their analysis, the phases and weights of their traces will be obtained, so as to later perform a selection mechanism. This allows us to evaluate the process that has similar behavior to the application, which is used to build the signature of the application. In this section, we will explain in more details our proposal.

4.1 Load Data

In order to obtain information about the behavior of the parallel applications, the instrumentation module is used. The MPI primitives will be saved as communication events and computational time to build one trace file per process. As is shown in Fig. 5, during the instrumentation, each process generates its trace file with the communication with the computation information involved in the process; thereby, the instrumentation module generates the same number of trace files as processes.

The load of the trace files is distributed using the same number of processes that the analyzer module has, as illustrated in Fig. 6.

4.2 Application Model

To create an abstract model of the application, all events must be sorted. Our proposal uses the concept of logical time defined by Lamport [19]. Lamport defined the relationship of precedence between two events a and b , where a occurs before b as the physical clock of the process a is smaller than the physical clock of the process b .

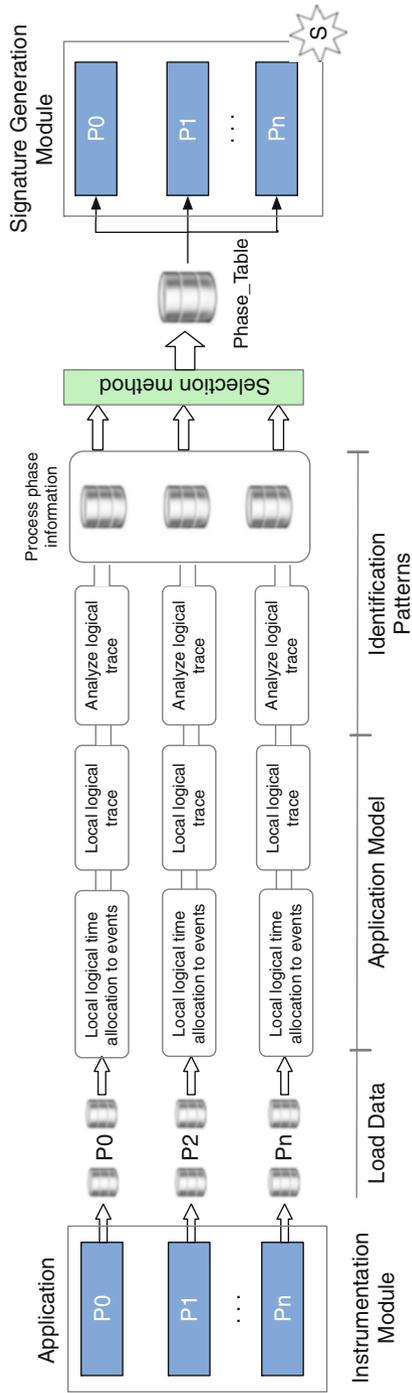


Fig. 5 Overview extension of parallel analysis to SPMD

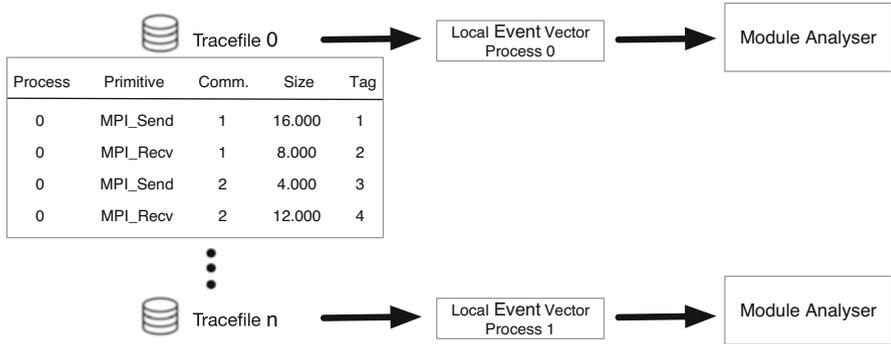


Fig. 6 Loads the trace data into the local event vector

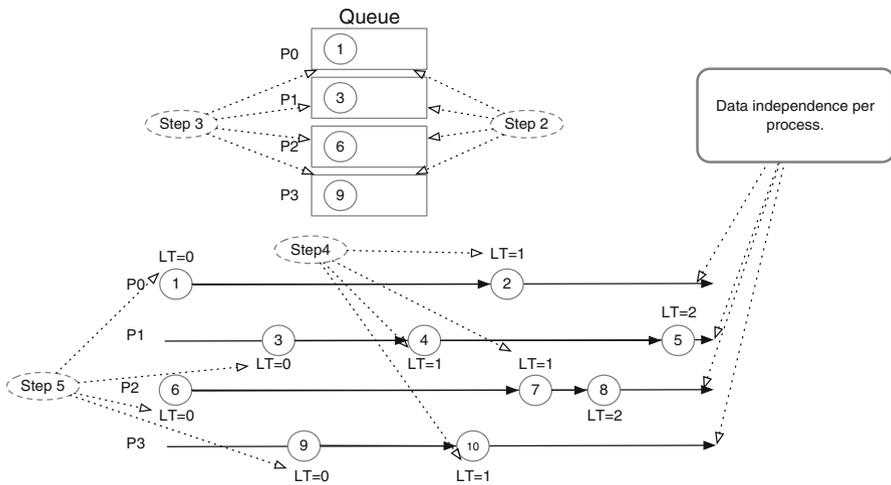


Fig. 7 Insertion of logical times

To assign the logical times (LT) in the EPAS proposal, the events of the logical trace of each process, obtained from the data loading stage, must be used. In our proposal, the logical times are built consecutively according to the events of each process. This is because each process is independent of the other, avoiding communication for synchronization. The LT assignment algorithm is illustrated in Fig. 7, where each process performs the following steps:

1. All events start with a logical time of zero.
2. A queue is created, and the first events are inserted.
3. The first event, CurrentEvent, is extracted from the queue.
4. The next consecutive event of the same process, ForwardEvent, is inserted in the queue.

5. You get the previous event, BackEvent, which has a smaller physical time than the CurrentEvent, where its logical time is assigned as follows:
 - (a) If there is no BackEvent event, CurrentEvent is assigned a logical time equal to 0 (CurrentEventLT=0).
 - (b) If there is a BackEvent event, the CurrentEvent logical time is the logical time of BackEvent + 1 (BackEventLT+1).
6. The procedure is completed when the queue is empty.

Our proposal differs from the implementation of the parallel analyzer [8] because we have designed a new model of the application where each process is independent of any other, allowing us to assign logical time, without replicating the communication pattern of the application to send the information of the logical times of each event, as well as using collective MPI to synchronize all processes. This new application model allows us to reduce the communication between processes, thus reducing the analyzer’s execution time.

When all the events of each process have their logical time and are ordered, we insert the emission events that the process has, within its logical trace. The logical trace is composed of a dimension that has as its size the maximum tick (ticks are defined as a unit of logical time), as is illustrated in Fig. 8. The value zero means that the event did not occur, and a value different from zero refers to the type of communication events that occurred.

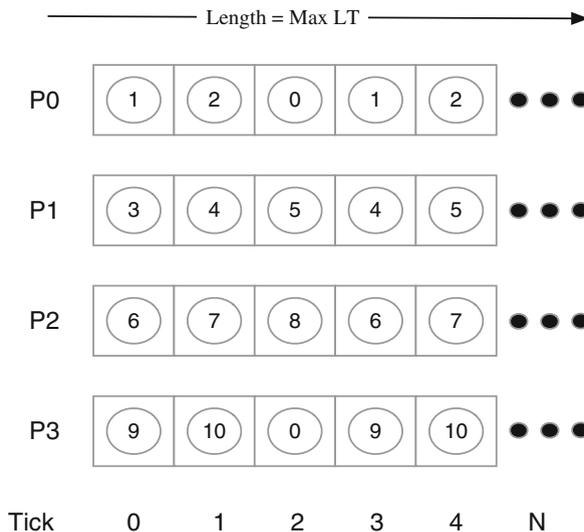


Fig. 8 Local logical trace

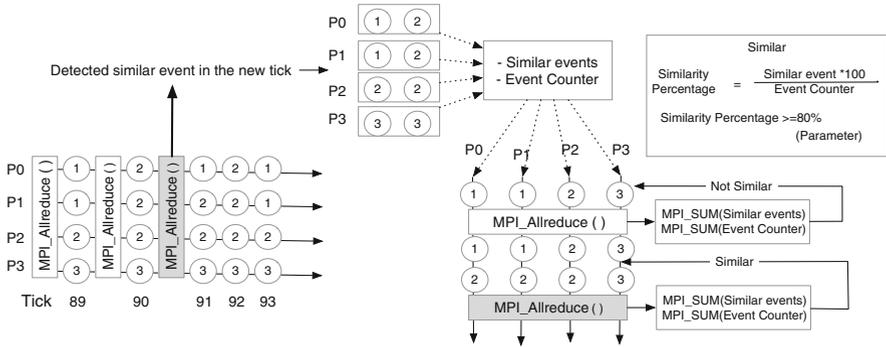


Fig. 9 Parallel Similarity Algorithm

4.3 Identification Pattern

In order to identify the phases of the parallel application, PAS2P extracts the information of each event directly from the logical trace. In the parallel version of the analyzer module, as in our proposal, each process has its own logic trace, as shown in Fig. 5.

The pattern identification method of the parallel analysis module compares all the events of a tick with another tick in memory-finding patterns of repeatability for all the processes, i.e., making vertical comparisons of the trace, as illustrated in Fig. 9. The phase extraction is performed using the similarity algorithm with specific criteria obtained by measuring the number of events and similar events.

In the method described above, it is necessary to use the communications between processes, as shown in Fig. 9, since the logical trace is distributed among all the processes. In order to perform the analysis, it is necessary to use a synchronization mechanism with the object of comparing events of the same logical time (tick), obtaining information that allows the use of the similarity algorithm to detect phases based on the type and volume of communication as well as the computed time.

Our proposal, illustrated in Fig. 10, aims to eliminate the use of communications between processes, analyzing the logical trace independently for each process, creating phases as small as possible. For each type of event, a phase is defined, which will be used to search for similarity in repeatability patterns.

Below are the necessary steps of our similarity proposal, illustrated in Fig. 10. It is important to note that this algorithm is performed using the nondependence of the processes in SPMD applications:

1. Startpoint and endpoint are created with the event of the first tick of the logical trace, storing it in a phase structure.
2. Each event of the phase structure is compared with the next tick of the logical trace, verifying that the phase exists, following these criteria:

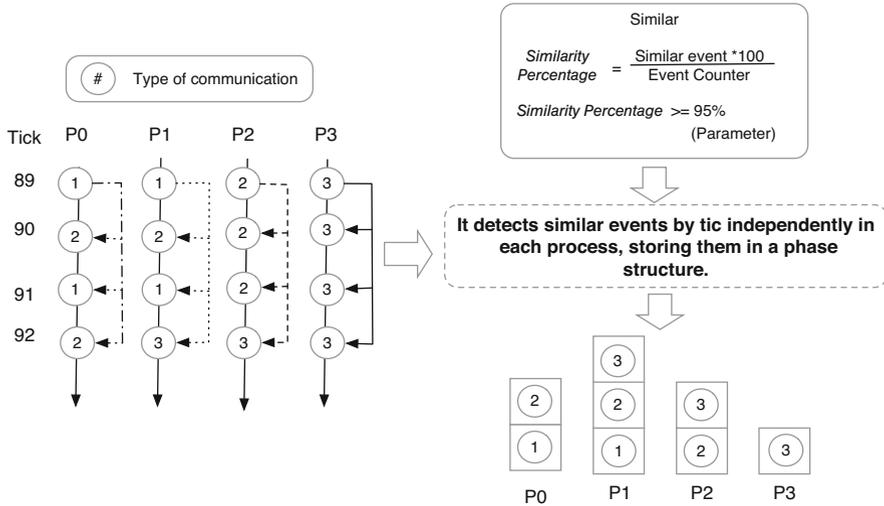


Fig. 10 Extension Parallel similarity algorithm

- (a) If the event does not occur with the same type of communication, the event is added to the phase structure by advancing one tick on the logical trace and returning to Step 2.
- (b) If the event occurs with the same type of communication, the following criteria must be met for a phase to exist:
 - i. The number of instructions between the two events must be similar (85% similarity or more).
 - ii. A phase is similar if the number of similar events is greater than or equal to 95% (configurable value) of the total number of events that the phase has.
 - If it is similar, the phase weight increases, and it advances one tick in the logic of the process.
 - If it is not similar, it is saved as a new phase in the phase structure, and it advances one tick in the logic of the process.

As illustrated in Fig. 10, each process generates measurements, related to the similarity of events and the number of events per phase, in order to obtain a percentage of similarity between the events of the same logical trace. In the case of our proposal, these values will be between 0 and 1, because the similarity algorithm compares phases that contain only one event.

Once the logical trace is fully traced using the pattern identification algorithm, as seen above, all its phases, together with their weights, are stored independently in a structure in the main memory.

The above structure contains the events that meet the similarity criteria (phase), along with their respective weight vectors. With this information, we can predict

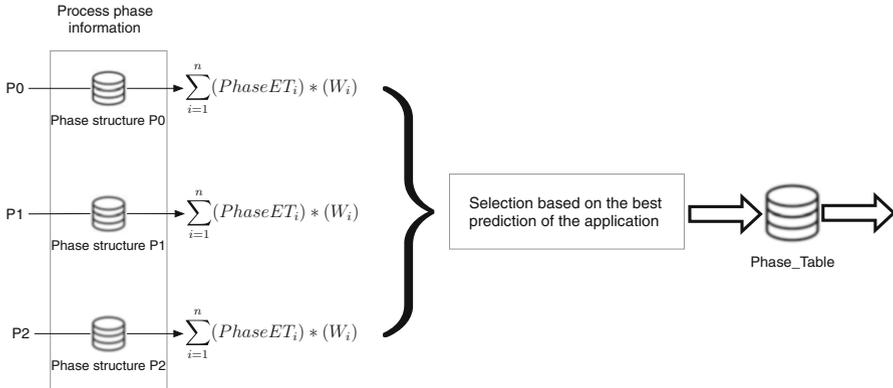


Fig. 11 Phase selection

in the base machine the execution time of the application by process p , as can be observed in Equation 1, where we multiply the execution time of each phase ($PhaseET_{b_i}$) by its weight ((W_{b_i})), obtaining a prediction of its execution time in process p :

$$PET_{b_p} = \sum_{i=1}^n (PhaseET_{b_i}) * (W_{b_i}) \tag{1}$$

In Fig. 11, we illustrate our proposal of phase selection using the prediction of the execution time (on the base machine) of the application, carried out by a process. We can say that a process can characterize the application because our proposal uses single program, multiple data (SPMD) [20] applications. Thus, we select the process that has the lowest percentage of error in the prediction of the execution time of the application, to generate the global phases file, Phase_Table.

The global phase file (Phase_Table) is the result of our analysis proposal. In this file, the phases and weights selected above are stored. Figure 12 shows an example of the content of the Phase_Table file, obtained when executing an application with 64 processes. It contains the information of the process selected in the module of extension of the parallel analysis to SPMD. The file shows the startpoint and endpoint of each phase. Each row of the table represents a phase. The third and fourth columns represent the phase ID and subphase, and the last column represents the phase weight.

The last stage is building the application signature; the process with ID=0 reads the global phase file (Phase_Table) and is communicated to all other processes using a collective MPI, as seen in Fig. 13. Each process instruments the application (binary), using the same library that we use to instrument the application. The library interacts with the application to detect relevant phases during the execution of the application and builds the signature.

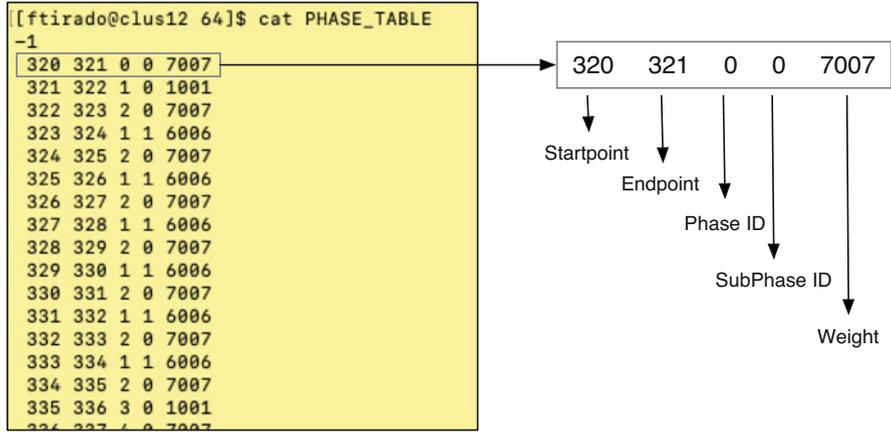


Fig. 12 Phase_Table

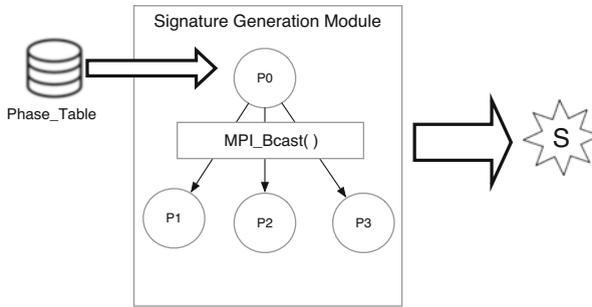


Fig. 13 Generate application signature

5 Experimental Results

In this section, we present the time required by our proposal to extend the parallel analysis to SPMD applied to PAS2P tools. The experimental methodology consists of running a set of applications and increasing the number of processes to validate our analysis proposal, as we scale the applications, compared to the parallel analysis version of PAS2P.

For the running environment, we use the Dell machine characterized in Table 1. The application SP from the NPB [9] was compiled for 36 to 256 processes, using class D as the workload with 1000 iterations. BT from the NPB [9] was compiled for 16 to 121 processes, using class C as workload with 5000 iterations. Finally, MiniMD [10] was compiled for 16 to 128 processes, using a problem size of $100 \times 100 \times 100$ with 5000 iterations.

We run the applications with PAS2P instrumentation to generate the data files. The version of the PAS2P parallel analyzer is compared with our EPAS proposal,

Table 1 Cluster characteristics

Cluster	Characteristics
DELL	AMD Opteron™ 6200 1.60 GHz, 8 nodes (512 cores), 64 GB RAM per node, Interconnection Gigabit Ethernet

comparing the time required to analyze the data from each application, increasing the number of processes. The size of the trace file is shown in the trace size column of Table 2, which represents the sum of all the trace files generated by each process of the application. The analysis of the trace file is proportional to the size of its trace files.

In the same Table 2, for the trace file analyzer time (TFAT) using the EPAS method, we obtain better performance in all cases compared to the parallel TFAT version. Here, we reach a gain of up to 11 times less than the parallel TFAT, due to the independent analysis model of each process, which avoids the use of communication for synchronization reasons. For the parallel analyzer, there is a considerable increase in the runtime from the 64 processes, which is because the Dell machine has 64 processes per node and starts using a network, which does not happen in our proposal because the analysis is independent of data between processes.

Also, Table 2 shows the prediction of the execution time (PET) to a different number of processes when executing the PAS2P tool with all its stages. We run the application instrumented in order to obtain the trace files and the execution time represented by the AET (application execution time) column. With this information, we calculate the percentage of error in the prediction of the execution time (PETE).

The results of the prediction times presented in the Table 2 show that our proposal has an average predictive quality of 97.37% compared to the parallel proposal, which has an average predictive quality of 96.25%. In addition, it is observed that the miniMD [10] application presents a better prediction using our EPAS (extension of the parallel analysis to SPMD) proposal, due to the topological application behavior that favors process analysis.

Finally, we present Fig. 14, which represents the execution time of the SP application analysis, using the parallel and extension of the parallel analysis to SPMD approaches. We observe a flattening of our proposal's curve as the number of processes increases, compared to the curve of the parallel analysis, which presents an accelerated rate of growth when increasing the number of processes.

6 Conclusion and Future Work

In this article, we present an improvement in the analysis of the PAS2P tool's trace file in SPMD applications, minimizing the use of communications, which allows us to increase the performance of the application analysis. We propose using all

Table 2 Analysis time of the trace file, parallel, and local approach

App.	Number processes	Parallel		Trace size (GB)	EPAS		Parallel approach		AET (s)
		EPAS TFAT (s)	TFAT (s)		approach PET (s)	approach PETE (%)	approach PET (s)	approach PETE(%)	
SP	36	2,03	6,25	0,27	4293,32	0,95	4284,78	0,75	4252,63
	64	5,15	15,89	0,66	3378,9	1,09	3359,54	0,52	3342,22
	121	5,59	61,44	1,74	2728,47	2,45	2680,06	0,69	2661,65
	256	10,81	119,35	5,60	1625,51	4,56	1632,39	4,96	1551,4
BT	16	4,30	12,39	0,33	6865,74	0,93	6856,94	0,80	6802,03
	36	8,88	25,99	1,1	3157,05	2,31	3123,11	1,25	3084,02
	64	14,17	45,07	2,6	1838,07	4,50	1797,09	2,33	1755,3
	121	19,21	163,11	6,7	1107,37	6,25	1088,21	4,60	1038,12
MiniMD	16	2,65	5,84	0,17	2848,77	1,42	2436,22	13,25	2808,45
	32	3,03	7,35	0,34	1419,04	0,4	1344,38	4,88	1413,34
	64	4,49	13,89	0,68	680,81	4,53	664,19	6,86	713,12
	128	6,08	32,86	1,34	407,84	2,19	382,62	4,08	398,89

TFAT trace file analyzer time, *EPAS* extension of the parallel analysis to SPMD, *PET* prediction execution time, *PETE* prediction execution time error, *AET* application execution time, *EPAS* extension of the parallel analysis to SPMD

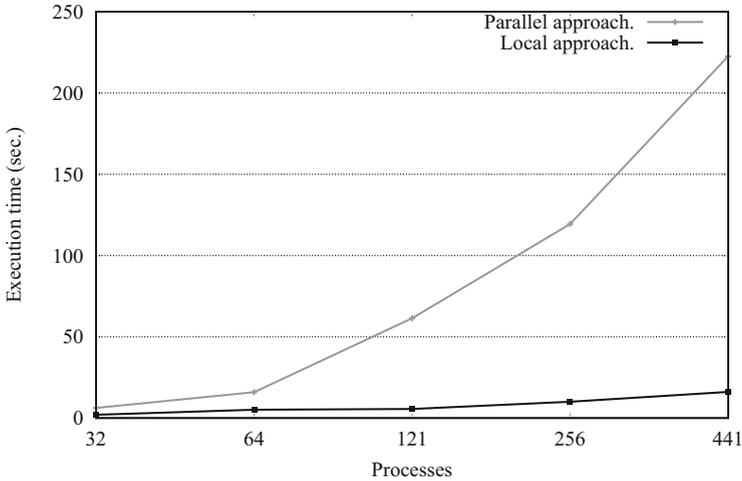


Fig. 14 Parallel and local approaches of the analyzer module in the SP application

the resources to run the application, performing an independent analysis for each process, and thus avoiding communications due to synchronization.

Our proposal of parallel analysis extension to SPMD managed to speed up the execution time in all the tests performed compared to the parallel analysis, keeping the prediction errors under 7%. This acceleration is seen in greater magnitude when the application is executed with a greater number of processes, due to the greater amount of communication for synchronization reasons.

As future work, we need to extend the characterization of the application, generating several signatures of the application, allowing us to analyze other types of scientific applications rather than SPMD.

Acknowledgments This paper is supported under contract TIN2017-84875-P (AEI/FEDER, UE) and partially funded by EUG.

References

1. N. Attig, P. Gibbon, T. Lippert, Trends in supercomputing: the European path to exascale. *Comput. Phys. Commun.* **182**(9), 2041–2046 (2011)
2. F. Wolf, B. Mohr, J. Dongarra, S. Moore, Automatic analysis of inefficiency patterns in parallel applications. *Concurrency Comput. Pract. Exp.* **19**(11), 1481–1496 (2007)
3. M. Geimer, P. Saviankou, A. Strube, Z. Szebenyi, F. Wolf, B.J.N. Wylie, Further improving the scalability of the Scalasca toolset, in *Proceedings of PARA 2010: State of the Art in Scientific and Parallel Computing, Part II: Minisymposium Scalable Tools for High Performance Computing*. Lecture Notes in Computer Science, vol. 7134 (2012), pp. 463–474
4. H. Brunst, H.-C. Hoppe, W.E. Nagel, M. Winkler, Performance optimization for large scale computing: the scalable vampir approach., in *ICCS* (2001), pp. 751–760

5. M. Gerndt, K. Frlinger, E. Kereku, Periscope: advanced techniques for performance analysis, in *PARCO*, vol. 33 (2005), pp. 15–26
6. J. Panadero, A. Wong, D. Rexachs, E. Luque, A tool for selecting the right target machine for parallel scientific applications, in *ICCS* (2013), pp. 1824–1833
7. A. Wong, D. Rexachs, E. Luque, Parallel application signature for performance analysis and prediction. *IEEE Trans. Parallel Distrib. Syst.* **26**(7), 2009–2019 (2015)
8. F. Tirado, A. Wong, D. Rexachs, E. Luque, Analyzing the data behavior of parallel application for extracting performance knowledge, in *2019 IEEE 21th International Conference on High Performance Computing and Communications* (IEEE, 2019). Accepted
9. D.H. Bailey, E. Barszcz, J.T. Barton, D.S. Browning, The nas parallel benchmarks, in *Proceedings of the 1991 ACM/IEEE Conference on Supercomputing*. Supercomputing'91 (1991), pp. 158–165
10. P. Crozier, S. Plimpton, minimd v. 1.0. Technical report, Sandia National Laboratories (2009)
11. J. Labarta, J. Gimenez, E. Martnez, P. Gonzlez, H. Servat, G. Llort, X. Aguilar, Scalability of tracing and visualization tools, in *PARCO* (2005), pp. 869–876
12. J. Labarta, New analysis techniques in the cepba-tools environment, in *Tools for High Performance Computing 2009* (Springer, Berlin/Heidelberg, 2010), pp. 125–143
13. T. Hilbrich, M. Schulz, B.R. de Supinski, M.S. Mller, Must: a scalable approach to runtime error detection in mpi programs, in *Parallel Tools Workshop* (2009), pp. 53–66
14. L. DeRose, B. Homer, D. Johnson, S. Kaufmann, H. Poxon, Cray performance analysis tools, in *Tools for High Performance Computing* (Springer, Berlin, 2008), pp. 191–199
15. S. Shende, A. Malony, G. Allen, J. Carver, S. Choi, T. Crick, M.R. Crusoe, Using TAU for performance evaluation of scientific software, in *Workshop on Sustainable Software for Science: Practice and Experiences*, vol. 1686 (2016)
16. S.S. Shende, A.D. Malony, A. Morris, Improving the scalability of performance evaluation tools, in *Proceedings of PARA 2010* (2010), pp. 441–451
17. A. Morari, M. Valero, HPC system software for regular and irregular parallel applications, in *2013 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum* (IEEE, 2013), pp. 2242–2245
18. D. Terpstra, H. Jagode, H. You, J. Dongarra, Collecting performance data with PAPI-C, in *Tools for High Performance Computing 2009* (Springer, Heidelberg, 2010), pp. 157–173
19. L. Lamport, The ordering of events in a distributed system. *Commun. ACM* **21**(7), 558–565 (1978)
20. F. Darema, The SPMD model: past, present and future, in *European Parallel Virtual Machine/Message Passing Interface Users' Group Meeting* (Springer, 2001), p. 1

Directive-Based Hybrid Parallel Power System Dynamic Simulation on Multi-core CPU and Many-Core GPU Architecture



Cong Wang, Shuangshuang Jin, and Yousu Chen

1 Introduction

The evolution of the clean energy economy and energy security demand requires an urgent modernization and continual expansion of the electric power grid. However, dynamically modeling the behaviors of such a large-scale complex transmission system requires computationally intensive time-domain simulations. Although the use of high-performance computing (HPC) techniques can greatly accelerate computation, adopting HPC-based applications in today's power utilities remains a big challenge. This is largely due to the reasons that developing HPC-based parallel applications to replace the legacy central processing units (CPU)-based sequential power system simulation tools requires significant programming efforts, and transitioning from the use of CPU-based commodity computer or workstation to HPC-based supercomputer is an expensive proposition. Currently, most of the commonly used power system commercial simulation tools are running on Windows-based platforms, such as GE PSLF Simulation Engine [1], Siemens PSS/E [2], DSATools [3], and PowerWorld Simulator [4]. They are widely used in today's power industry due to mature sequential algorithm-based approaches that have been developed for decades and user-friendly interface for power engineers to fulfill the system control and operation needs. Nevertheless, there is little clue on how the advanced parallel techniques could be well applied to further improve the computational performance of these fine-tuned algorithms. Their sequential execution performance of a certain application on large-scale power systems heavily

C. Wang (✉) · S. Jin
Clemson University, North Charleston, SC, USA
e-mail: cong2@clemson.edu; cong2@g.clemson.edu; jin6@clemson.edu

Y. Chen
Pacific Northwest National Laboratory, Richland, WA, USA

depends on the CPU capabilities of the workstation they reside in, resulting in insufficient real-time responses and diagnosis.

General-purpose computing on graphics processing units (GPUs), in combination with CPU-based computing, has great potential to lower this barrier due to its superior floating-point acceleration performance and cost-effective architecture. CPU+GPU-based applications, which accommodate most legacy sequential algorithms through directive-enabled parallelism, use affordable hardware, exhibit commendable computational performance, and, with high energy efficiency, are an ideal point of departure to begin this HPC revolution.

In this work, an Open Multi-Processing (OpenMP)-based [5] and Open Accelerators (OpenACC)-based [6] hybrid parallel power system dynamic simulation is developed to showcase the advantage of leveraging multi-core CPU and many-core GPU computing to enable real-time power system simulations for advanced grid analytics with minimum modification to the existing sequential data flows and algorithms. Section 2 provides a brief overview of the HPC trends in large-scale power grid analysis as well as an introduction to power system dynamic simulation. Section 3 illustrates the proposed directive-based hybrid approach and how it is used to decouple the time-domain problem of differential and algebraic equations (DAEs) for dynamic simulation. Section 4 presents the computational performance of the implementation and a comparison against single-CPU serial code and multi-core CPU OpenMP-only implementations using two power system test cases (IEEE 145-bus and Polish 3120-bus). Section 5 concludes the paper with an overview of future work.

2 Background

2.1 HPC in Power Grid

As the demand for computing resources continues to increase, HPC is gaining more and more attention in facilitating accelerated modeling and simulation in modern electric power grid. Khaitan and Gupta [7] summarizes a variety of HPC-related applications, e.g., distributed parallel power system simulation, cluster computing for Energy Management System (EMS), etc., to boost the performance of the complex system. Besides, several HPC-enabled power system simulators or frameworks are developed to facilitate the modeling and simulation, such as GridLAB-D [8], which exploits multi-threading techniques to achieve synchronized state evaluations of all components in a power distribution system, and GridPACKTM [9] which endeavors to substantially reducing the complexity of coding effort for parallel computing while providing efficient and scalable software solutions for power grid simulations.

In recent years, CPU+GPU computing is emerging to provide accelerated computations to various scientific and engineering research problems including power systems [10–14]. In the context of a hybrid CPU+GPU computing framework, GPU

consisting of thousands of smaller but efficient cores for parallel computing acts as the acceleration engine, which is in charge of high-level parallel kernels with data-intensive computation to optimize the execution throughput of a massive number of threads utilizing GPU's highly parallel structure and exceptional floating-point computing efficiency [15]. Conversely, CPU manages overall data flow and other noncritical computing, branch condition, data transfer, and I/O operations in the typical serial fashion through the use of its sophisticated control logic and large cache memories to reduce the instruction and data access latency for large complex applications. As a result, in most power system applications, the data parsing, network configuration, and I/O transition are serial processing tasks performed in CPU, while the most computationally demanding tasks such as matrix vector operations and linear/nonlinear system calculations, are off-loaded to GPU for acceleration.

2.2 Dynamic Simulation for Transient Stability Analysis

Dynamic simulation, as a critical function for operational decision-making, involves specific power system component modeling and massive time domain numerical computation. It is used to evaluate system's transient trajectories when there is a system disturbance (fault), e.g., a sudden change in generator or load, or a network short circuit followed by protective branch switching operations, etc. Modeling the system dynamics and network requires computationally intensive time-domain solution of numerous DAEs as shown in (1)

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{0} = \mathbf{g}(\mathbf{x}, \mathbf{u}) \end{cases} \quad (1)$$

where the vector \mathbf{x} represents dynamic state variables such as generator rotor angles and speeds and the vector \mathbf{u} represents algebraic variables such as the network bus voltage magnitudes and phase angles and real and imaginary parts of the bus voltage [16]. The intensive computation involved in these processes makes deriving solutions to these equations a dominant time-consuming limiting factor to meet the real-time constraints that characterize the dynamic assessment of online security. The equations of motion for an individual generator i in the complex system could be represented by (2) if a classical generator model is used in the dynamic simulation

$$\begin{cases} \frac{dw_i}{dt} = \frac{w_s}{2H_i}(P_{mi} - P_{ei} - D_i(w_i - w_s)) \\ \frac{d\theta_i}{dt} = w_i - w_s \end{cases} \quad (2)$$

for generator i , H_i is the inertia constant, w_i is the speed, w_s is the synchronous speed, P_{mi} and P_{ei} are the mechanical power input and active power at the air gap,

D_i is the damping coefficient, and θ_i is the angular position of the rotor in the electrical radians with respect to synchronously rotating reference.

The essential steps of solving dynamic modeling in the simulation are as follows [17]:

1. The pre-fault condition power flow solution.
2. Constant impedance conversion.
3. Full \mathbf{Y} matrix (nodal admittance matrix) formation.
4. Reduced \mathbf{Y} matrix (\mathbf{Y}') formation.
5. Power-angle equations formulation for pre-fault, on-fault, and post-fault conditions.
6. Machine swing equations formation.
7. Numerical integration.

In Step 3, the algebraic equations in (1) can be represented by (3)

$$\mathbf{Y}\mathbf{V} = \mathbf{I} \quad (3)$$

The vector of current injections \mathbf{I} at each bus is expressed as the production of full \mathbf{Y} bus and the vector of bus voltages \mathbf{V} . To simplify the complexity of the matrix and achieve the best matrix operation performance, for a power system with classical model in (2) and constant impedance load, full \mathbf{Y} matrix can be reduced to only contain generator internal buses, \mathbf{Y}' . According to [18] and [19], (4) represents the logic of acquiring reduced nodal admittance matrix in Step 4:

$$\mathbf{Y}' = \mathbf{Y}_{mm} - \mathbf{Y}_{mn}\mathbf{Y}_{nn}^{-1}\mathbf{Y}_{nm} \quad (4)$$

where m is the number of generators and n is the total number of buses in the power system. \mathbf{Y}_{mm} is the matrix storing generators' resistance and reactance, \mathbf{Y}_{mn} are the links between generator internal buses and terminal buses, \mathbf{Y}_{nn} contains constant load impedance and generator transient impedance, and \mathbf{Y}_{nm} is the transpose of \mathbf{Y}_{mn} . Thus, the algebraic equations can be modified to

$$\mathbf{Y}'\mathbf{V}' = \mathbf{I}', \quad (5)$$

where \mathbf{I}' is the current injection and \mathbf{V}' is the internal voltages of generators.

3 Proposed Approach

3.1 Algorithm Decoupling

To solve the DAEs of dynamic simulation, the differential equation set in (1) needs to be first discretized into algebraic equations, which are then lumped with the original algebraic equations to be solved. A widely used method is explicit integration such as the modified Euler [20] method in (6):

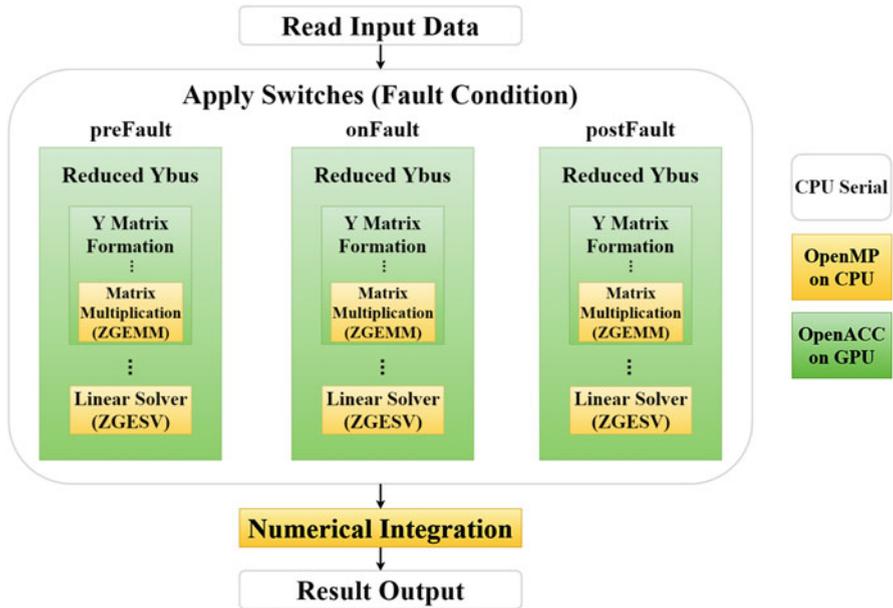


Fig. 1 Algorithm decoupling and CPU+GPU implementation

$$\begin{cases} \text{Predict} : y_{i+1} = y_i + hf(t_i, y_i) \\ \text{Update} : y_{i+1} = y_i + \frac{h}{2}[f(t_i, y_i) + f(t_{i+1}, y_{i+1})] \end{cases} \quad (6)$$

Given the input variables, h is the step size, $f(t, y) = y'$ and $y(t_0) = \alpha$, where α is the initial value of y at time step t_0 , $t_{i+1} = t_i + h$. At each integration time step i , the DAEs are solved twice alternately to update the system state. The differential equations can inherently be assigned to different computing threads to be processed in parallel; the algebraic equation involving significant matrix operations (e.g., the \mathbf{Y} matrix formation) and linear system solutions (e.g., the network coupling equations) can also be accelerated through parallel implementation. Figure 1 depicts the algorithm decoupling and parallelism implementation in this work based on the sequential program in Power System Toolbox (PST) [21].

3.2 Parallel Implementation

The parallel dynamic simulation application on the hybrid CPU+GPU architecture is developed in Fortran 90 using OpenMP, an Application Programming Interface (API) that supports multi-platform shared memory multiprocessing programming, and OpenACC, a user-driven directive-based performance-portable parallel programming model for GPU computing. The hybrid approach enables maximum

```

!$acc data copyout(chrgfull(1:nbrch,1:nbrch),yyfull(1:nbrch,1:nbrch))
copyin(yy(1:nbrch,1),chrg(1:nbrch,1))
!$acc parallel loop private(i)
    DO i=1,nbrch,1
        chrgfull(i,i)=chrg(i,1)
        yyfull(i,i)=yy(i,1)
    END DO

!$acc parallel loop private(i)
    DO i=1,nbrch,1
        ...
    END DO
!$acc end data

```

Fig. 2 An example of OpenACC implementation in the hybrid program

flexibility of parallelism leveraging both multi-core CPU's linear algebra libraries through OpenMP directives and many-core GPU's massive computing capability through OpenACC directives. As shown in Fig. 1, other than a few noncritical computing components (e.g., switching conditions, I/O operations) which remain sequential in the hybrid program, all the other time-critical computational components (e.g., matrix multiplication, numerical integration, linear system solving) are enabled by either OpenMP or OpenACC for acceleration.

An example of implementing OpenACC directives on matrix manipulation during the \mathbf{Y} matrix formation is shown in Fig. 2 to illustrate the simplicity of parallel implementation without changing the legacy sequential program. The compiler directive `!$acc parallel loop` explicitly instructs the compiler to launch the GPU kernel for parallelism. `!$acc data` defines a data region in which matrices on GPU will remain and can be shared among all kernels until the end of data region. The data clauses `copyin` and `copyout` allocate memory on GPU and copy data from CPU to GPU when entering region and from GPU to the CPU when exiting region.

To speed up the computational time in Steps 5 to 7 of the simulation, an OpenMP-based parallel modified Euler method is implemented, with three simple OpenMP directives of “do” loops as shown in Fig. 3. Multiple threads are spawned at each of the loops to execute simultaneous instructions on different generators. The nature of both OpenMP and OpenACC simplifies the HPC coding effort to increase computational speed without compromising the accuracy of original algorithms.

A thread-safe version Basic Linear Algebra Subprograms (BLAS) [22] and Linear Algebra PACKage (LAPACK) [23] are utilized to perform the complex number matrix-matrix multiplication and solve the complex system of linear equations through OpenMP Fortran subroutines such as ZGEMM [24] and ZGESV [25]. The thread-safe implementation of BLAS and LAPACK allows ZGEMM and ZGESV to perform concurrently on different data sets (e.g., different block-wise data elements in a matrix for ZGEMM) or independent problems (e.g., different group of right-hand side vectors of a linear system for ZGESV). Thus, the total computational time is reduced by multiple threads running on multi-core CPUs [26], enabling a guaranteed performance boost in computation without rewriting the legacy sequential program.

```

!$OMP DO PRIVATE(k)
  DO k=1,ngen,1
    pmech(k,S_Steps+1)=pmech(k,S_Steps)
    CALL mac_em1(k,S_Steps)
  END DO
!$OMP END DO

!$OMP DO PRIVATE(k)
  DO k=1,ngen,1
    CALL i_simu_innerloop(k,S_Steps,flagF1)
    CALL mac_em2(k,S_Steps)
    !Eular prediction for next-timestep generator angle and speed.
    mac_ang(k,S_Steps+1)=mac_ang(k,S_Steps)+h_sol1*dmac_ang(k,S_Steps)
    mac_spd(k,S_Steps+1)=mac_spd(k,S_Steps)+h_sol1*dmac_spd(k,S_Steps)
    edprime(k,S_Steps+1)=edprime(k,S_Steps)
    eqprime(k,S_Steps+1)=eqprime(k,S_Steps)
    CALL mac_em1(k,S_Steps+1)
  END DO
!$OMP END DO

!$OMP DO PRIVATE(k)
  DO k=1,ngen,1
    CALL i_simu_innerloop(k,S_Steps+1,flagF2)
    CALL mac_em2(k,S_Steps+1)
    !Eular correction for next-timestep generator angle and speed.
    mac_ang(k,S_Steps+1)=mac_ang(k,S_Steps)+h_sol2*(dmac_ang(k,S_Steps)+&
      dmac_ang(k,S_Steps+1))/2.0
    mac_spd(k,S_Steps+1)=mac_spd(k,S_Steps)+h_sol2*(dmac_spd(k,S_Steps)+&
      dmac_spd(k,S_Steps+1))/2.0
  END DO
!$OMP END DO

```

Fig. 3 OpenMP-enabled modified Euler integration method

3.3 System Configuration

The working code is implemented on NVIDIA Tesla GPUs residing on Clemson University's supercomputing facilities Palmetto Cluster [27], which is a Linux-based environment comprised of 2,021 compute nodes (totaling 23,072 CPU cores) including 386 nodes equipped with NVIDIA Tesla GPUs. One computing node consisting of 16 CPU cores with 30 GB memory and 1 NVIDIA Tesla V100 16 GB GPU [28] is requested to perform the task. The GPU is composed of 6 GPU processing clusters (GPCs) and 8 512-bit memory controllers. Each GPC has 7 texture processing clusters (TPCs) and 14 SMs, and each SM wraps 64 FP32 cores and INT32 cores, 32 FP64 cores, 8 tensor cores, and 4 texture units. To ensure the code was running on the same platform, instead of using GNU compiler collection (GCC), PGI compiler version 19.4 [29], which has the ability to enable parallel interfaces such as OpenMP, OpenACC, and MPI, is selected.

4 Results and Analysis

To test the performance of the proposed algorithm, two widely acceptable power system test cases are selected to check its accuracy and scalability. The standard Siemens PTI format [30] representing power grid systems is adapted as the input to our hybrid dynamic simulation package.

4.1 Testing Cases

The IEEE 145-bus-50-generator system (IEEE145b) and a Polish 3120-bus-93-generator system (Polish3120b), which are converted from MATPOWER [31], an open-source tool for electric power system simulation and optimization, are selected as the test cases. For both cases, the simulation time is 30 s with a time step of 0.01 s. A fault is applied at bus 6 at the 3rd second and cleared at the 3.05 s to mimic a system disturbance and the relevant generators' dynamics (e.g., mechanical angles and speeds, power angles and speeds) as a response to the disturbance.

4.2 Performance of Scalability

The scalability of a parallel application is bounded by the non-parallelization portion of the program and influenced by the computation-to-communication ratio, i.e., computation intensity vs. communication overhead. As the number of computing threads increases, the computation time is expected to decrease until the capability is limited by the problem size when a certain number of computing threads are reached.

As a result, the larger Polish3120b system is expected to have a better scalability over the medium IEEE145b system as reflected in Figs. 4 and 5. The x-axis shows sequential CPU run (CPU serial), and the hybrid runs with different number of threads (ACC+number of OpenMP threads). For the IEEE145b system, saturation of computational performance starts as early as 2 threads (ACC+2), whereas for the Polish3120b system, it achieves performance gains proportionally until 40 threads (ACC+40) are used.

The speedup performance of the hybrid program is obtained by comparing its computational times against the single-core CPU serial run and multi-core CPUs parallel run with various number of OpenMP threads. When the number of OpenMP threads is set to 1, it is basically an OpenACC-only run (ACC+1).

For the IEEE145b system, as shown in Fig. 4, the CPU serial run takes 0.56 s, while the OpenACC-only run takes 0.16 s. The OpenMP+OpenACC run achieves its best performance at 0.13 s using 2 OpenMP threads, a speedup of 4.3 times over the CPU serial run. No significant speedup gains can be achieved beyond 2 threads due to the aforementioned saturation constrained by the problem size.

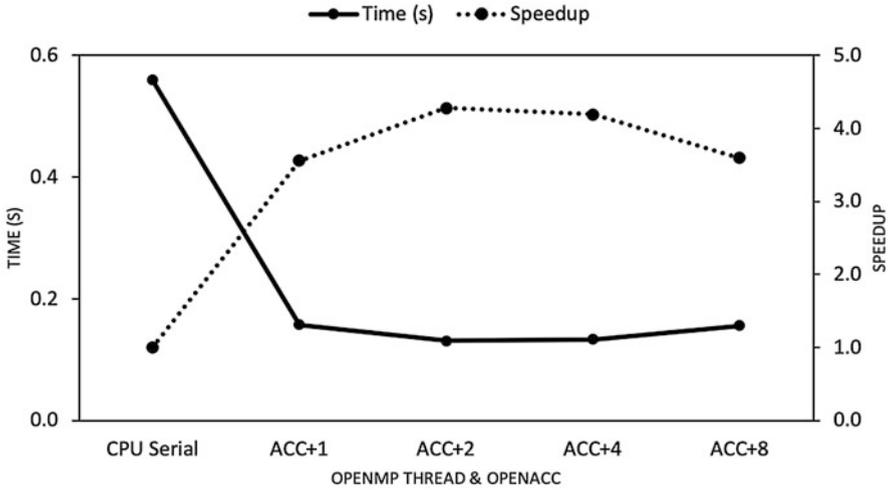


Fig. 4 The computational performance of parallel dynamic simulation on IEEE145b system

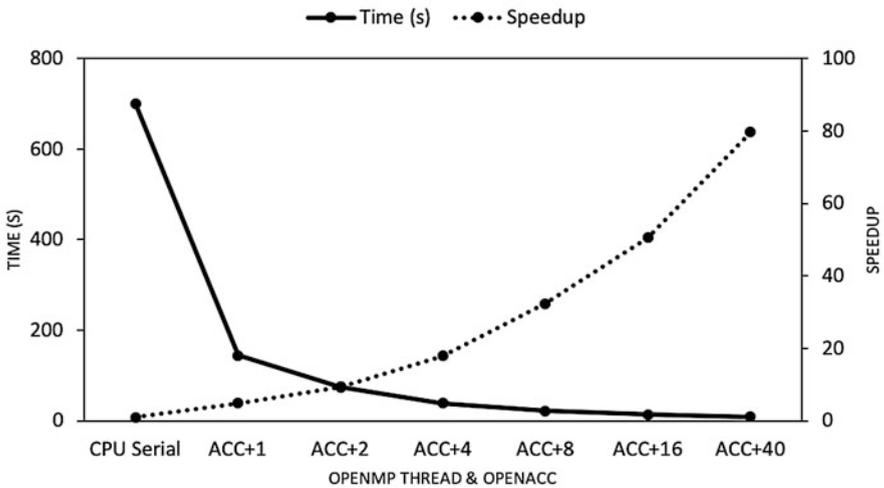


Fig. 5 The computational performance of parallel dynamic simulation on Polish3120b system

For the Polish3120b system, the computational intensity increases dramatically due to the non-optimized development in large matrix operations and linear equation solver implemented based on the legacy sequential algorithm. The CPU serial run takes over 700 s to complete a 30-s simulation, while the OpenACC-Only run takes 144.4 s, four times faster. When 40 threads are used with OpenMP+OpenACC, the best performance is achieved with a speedup of 79.7 and a simulation time of 8.8 s, over 20 s ahead of real time.

4.3 Advantage over OpenMP-Only Implementation

To further evaluate the efficiency of the hybrid implementation, a pure OpenMP-based dynamic simulation (OpenMP-Only) is also developed to exclude the GPU impact for bench-marking. Instead of loading computational intensive components onto GPU as depicted in Fig. 1, their parallelisms are enabled by OpenMP directives alternatively.

Figure 6 compares the computational time of the hybrid OpenMP+OpenACC run versus OpenMP-only with different number of threads for the Polish3120b case. Considering the peak performance, the hybrid program (8.8 s with 40 threads) beats the OpenMP-only run (53.1 s with 16 threads) by six times. Table 1 summarizes the comparisons on the shortest execution time across all different implementations described in this section. These results demonstrate that, with the addition of OpenACC and the associated small programming efforts of OpenACC directives, the hybrid implementation outperforms the OpenMP-only one for all cases, making it an efficient and easily adopted technique to be accepted by commercial vendors, utilities, and power system engineers.

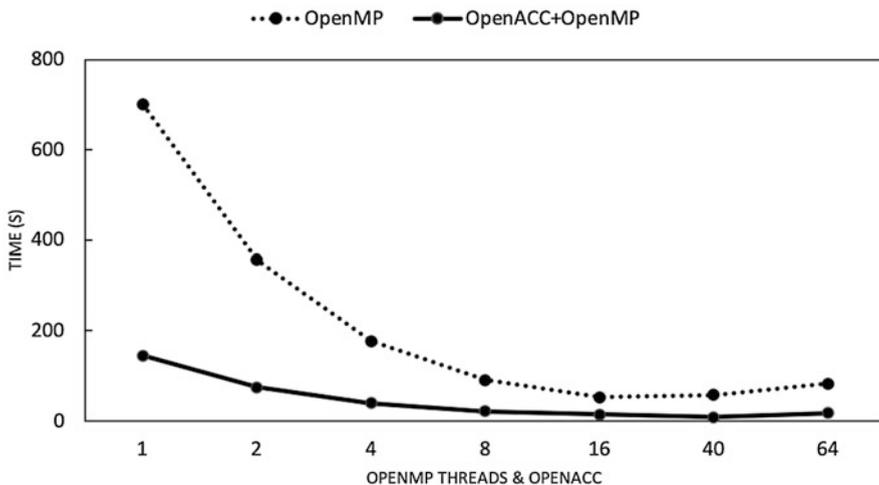


Fig. 6 Comparison of computational time between OpenMP+OpenACC and OpenMP-only implementations for Polish3120b cases

Table 1 Comparison on best performance across all different implementations

System	CPU serial	OpenACC-only	OpenMP-only	Hybrid
IEEE145b	0.56 s	0.16 s	0.18 s	0.13 s
Polish3120b	701.2 s	144.4 s	53.1 s	8.8 s

5 Conclusions and Future Work

This paper shows how efficiently the legacy serial program can be escalated to a high performance one using OpenMP+OpenACC-based hybrid parallel implementation to significantly increase computational speed of power system dynamic simulation without compromising the model accuracy. This hybrid architecture provides a simple and effective solution to foster the immediate use of advanced computing techniques in the power industry. The implementation technique can be easily extended to other power system applications to improve their computational performance as a whole. This improvement could dramatically enhance current situational awareness practice and decision-making process, making us in a great shape when facing increased grid complexities.

Besides extending the proposed computing framework to other power system applications, the future work includes optimizing OpenACC code to maximize the usage of GPU and transferring OpenMP-enabled matrix multiplication and linear equation solving into OpenACC or CUDA-optimized GPU code. We will also leverage the graphics interoperability features of GPU computing to realize seamless visualization on power system modeling and simulation to provide better visual analytics for decision-making.

References

1. GE PSLE, <https://www.geenergyconsulting.com/practicearea/software-products/pslf>. Accessed 10 Jun 2020
2. PSS/E Product Brochure, Siemens (2017)
3. DSATools, <https://www.dsatools.com/>. Accessed 23 Mar 2020
4. PowerWorld Simulator, <https://www.powerworld.com/>. Accessed 10 Jun 2020
5. OpenMP, <http://en.wikipedia.org/wiki/OpenMP>. Accessed 10 Jun 2020
6. OpenACC, <https://www.openacc.org/>. Accessed 18 May 2020
7. Khaitan S, Gupta A, High performance computing in power and energy systems (Springer, Berlin, 2014)
8. S. Jin, D.P. Chassin, Thread group multithreading: accelerating the computation of an agent-based power system modeling and simulation tool – C GridLAB-D, in *2014 47th Hawaii International Conference on System Sciences*, Waikoloa (2014), pp. 2536–2545
9. B. Palmer et al., GridPACK: a framework for developing power grid simulations on high performance computing platforms, in *2014 Fourth International Workshop on Domain-Specific Languages and High-Level Frameworks for High Performance Computing*, New Orleans (2014), pp. 68–77
10. V. Jalili-Marandi, V. Dinavahi, SIMD-based large-scale transient stability simulation on the graphics processing unit. *IEEE Trans. Power Syst.* **25**(3), 1589–1599 (2010)
11. D. Chen, H. Jiang, Y. Li, D. Xu, A two-layered parallel static security assessment for large-scale grids based on GPU. *IEEE Trans. Smart Grid* **8**(3), 1396–1405 (2017)
12. G. Zhou, Y. Feng, R. Bo, L. Chien, X. Zhang, Y. Lang, Y. Jia, Z. Chen, GPU accelerated batch-ACPF solution for N-1 static security analysis. *IEEE Trans. Smart Grid* **8**(3), 1406–1416 (2017)

13. D. Su, GPU accelerated algorithm for online probabilistic power flow. *IEEE Trans. Power Syst.* **33**(1), 1132–1135 (2018)
14. J. Greathouse, M. Daga, Efficient sparse matrix-vector multiplication on GPUs using the CSR storage format, in *SC14 International Conference for High Performance Computing, Networking, Storage and Analysis*, New Orleans (2014), pp. 769–780
15. D. Kirk, W. Hwu, *Programming Massively Parallel Processors A Hands-on Approach*, Morgan Kaufmann (2013)
16. P. Kundur, N.J. Balu, M.G. Lauby (eds.), *Power System Stability and Control* (McGraw-Hill, New York, 1994)
17. S. Jin, Y. Chen, D. Wu, R. Diao, Z. Huang, Implementation of parallel dynamic simulation on shared-memory vs. distributed-memory environments, in *IFAC* (2015), pp. 221–226
18. S. Jin, Z. Huang, R. Diao, D. Wu, Y. Chen, Comparative implementation of high performance computing for power system dynamic simulations. *IEEE Trans. Smart Grid* **8**(3), 1387–1395 (2017)
19. P.M. Anderson, A.A. Fouad, *Power System Control and Stability*, ed. by M.E. El-Hawary, 2nd edn. (Wiley, Piscataway Township, 2003)
20. K.A. Atkinson, *An Introduction to Numerical Analysis*, 2nd edn. (Wiley, New York). ISBN 978-0-471-50023-0
21. Power System Toolbox Version 3.0. <https://www.ecse.rpi.edu/~chowj/PSTMan.pdf>. Accessed 10 Mar 2020
22. L. Blackford, A. Petitet, K. Remington, R. Whaley, J. Demmel, An updated set of basic linear algebra subprograms (BLAS). *ACM Trans. Math. Softw.* **28**(2), 35–51 (2002)
23. E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra et al., *LAPACK Users' Guide*, 3rd edn. (Society for Industrial and Applied Mathematics, Philadelphia, 1999)
24. ZGEMM, <https://docs.oracle.com/cd/E19422-01/819-3691/zgemm.html>. Accessed 10 Mar 2020
25. ZGESV, <https://docs.oracle.com/cd/E19422-01/819-3691/zgesv.html>. Accessed 10 Mar 2020
26. FUJITSU, BLAS LAPACK User's Guide, <http://www.lahey.com/docs/blaseman.pdf>. Accessed 10 Mar 2020
27. Palmetto, <https://www.palmetto.clemson.edu/palmetto/> Accessed 18 May 2020
28. NVIDIA, NVIDIA Tesla V100 GPU Architecture Whitepaper, WP-08608-001 (2017)
29. NVIDIA, PGI Compilers and Tools User Guide for OpenPOWER CPUs (2019)
30. PTI Power Flow Format, <https://labs.ece.uw.edu/pstca/formats/pti.txt> Accessed 10 Mar 2020
31. R.D. Zimmerman, C.E. Murillo-Sanchez, R.J. Thomas, MATPOWER: steady-state operations, planning and analysis tools for power systems research and education. *IEEE Trans. Power Syst.* **26**(1), 12–19 (2011)

Parallel Computation of Gröbner Bases on a Graphics Processing Unit



Mark Hojnacki, Andrew Leeseberg, Jack O’Shaughnessy, Michael Dauchy, Alan Hylton, Leah Gold, and Janche Sang

1 Introduction

A Gröbner basis is a set of multivariate polynomials that can be derived from another set of polynomials over a finite field. It shares the same roots as its derivation but provides desirable properties that allow it to be used in algorithms more effectively [1]. In this way a Gröbner basis has many real-world applications including geometry, robotics, and computational algebra. However, given that a system can have many variables up to any degree, the method to compute a Gröbner basis can become difficult.

The first algorithm to compute Gröbner bases was released by Bruno Buchberger in 1965. The algorithm he introduced, aptly named Buchberger’s Algorithm, uses a criterion to test for completion of the Gröbner basis and S -polynomials to derive new polynomials [1]. The algorithm has been proven to compute a Gröbner basis, however, it can take a long time. In 1999, Jean-Charles Faugère released a new algorithm to compute Gröbner bases, called F4 [2]. This algorithm builds off the work of Buchberger, hoping to reduce the number of computations needed before the algorithm completes. It does this through the careful selection of the new

M. Hojnacki · A. Leeseberg · J. O’Shaughnessy · M. Dauchy · J. Sang (✉)
Department of Electrical Engineering & Computer Science, Cleveland State University,
Cleveland, OH, USA
e-mail: sang@eecs.csuohio.edu

A. Hylton
Space Communications and Navigation, NASA Glenn Research Center, Cleveland, OH, USA
e-mail: alan.g.hylton@nasa.gov

L. Gold
Department of Mathematics, Cleveland State University, Cleveland, OH, USA
e-mail: l.gold33@csuohio.edu

polynomial and heavy use of matrix reductions [2]. Faugère also created the F4 and F5 algorithm with new criteria to determine a Gröbner basis [1, 2].

Over time Faugère and others have created variants based off of Faugère’s F4 and F5 algorithms, including F4/5 and F5C [3, 4]. While there are various differences between each of the implementations the main purpose of each is to cut out a large chunk of unnecessary computations from the Buchberger algorithm. There are many cases in the Buchberger algorithm where polynomials are reduced to zero and are therefore redundant [2]. F4 preformats the data using sparse linear algebra [1]. F5 uses a series of criteria that Faugère refers to as the “signature” polynomials that would be reduced to zero and removes them before computation. F4/5 is simply a combination of the preformat from F4 combined with the “signature” polynomial selection from F5 [3]. F5C is a version of the F5 algorithm created by Christian Eder and John Perry that computes only the unique reduced Gröbner basis for the system and uses that for the pair selection [4].

The goal of this project was to utilize an NVIDIA GPU to accelerate parallel operations of a Gröbner basis algorithm and create an open-source repository to facilitate further development of Gröbner basis calculations. The rest of this paper is organized as follows: Background research for the project, description of the design methodology, implementation process and experiences, experimental results, and a brief conclusion and future work.

2 Background

Gröbner bases are difficult and time-consuming to compute due to their exponential complexity and growth factors. That is why a graphics processing unit (GPU) was utilized in this project to help speed up the computations. Typical Central Processing Units (CPUs) only have a few cores dedicated to general purpose computing tasks. On the other hand, GPUs are specialized microprocessors with hundreds or thousands of cores that are purpose built for parallel computations [5]. Utilizing these parallel processing cores can provide much higher data computational throughput than that of a CPU. The architectures of both a CPU and GPU can be seen in Fig. 1. Note that a CPU only has a few cores, a larger cache, and the main system memory. On the other hand, the GPU (different GPUs have different core counts and architectures) boasts over one hundred cores and shared memory for the cores to share. Given the right optimizations, applications such as computer graphics, data science, machine learning, weather simulations, and even chemistry computations can see large performance boosts by using a GPU [6, 7].

As previously mentioned, there are algorithms that benefit from the usage of linear algebra to do multiple calculations at the same time. Specifically, any matrix-based Gröbner algorithm typically will require many Gaussian eliminations to be computed, depending on the size of the input. Row echelon (REF) and reduced row echelon form (RREF) computations for matrices will be the primary algorithm that the GPU will focus on. NVIDIA offers coding libraries for its CUDA architecture

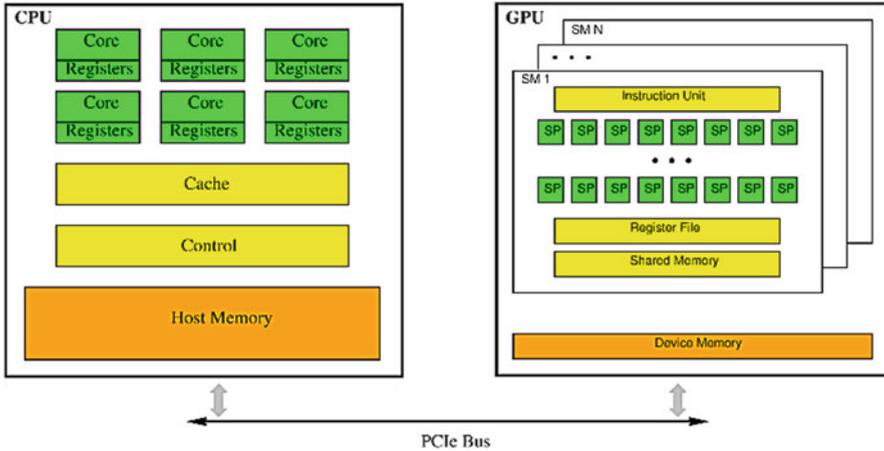


Fig. 1 CPU and GPU architecture

to simplify GPU programming with their graphics cards [5, 8, 9]. CUDA is NVIDIA’s parallel computing programming model for their graphics cards and CUDA programs can be coded in C, C++, Fortran, and other languages [10]. NVIDIA’s Application Program Interface (API) libraries such as cuBLAS and cuSPARSE are highly optimized to run on NVIDIA GPUs and thus were chosen to be used for this project. The cuBLAS library focuses on Basic Linear Algebra Subprograms [8]. These operations include vector, vector–matrix, and matrix–matrix operations on top of the CUDA runtime. The cuSPARSE library provides optimized operations for sparse matrices [9]. Sparse matrices are those with less nonzero entries than that of zero entries. Using one or both of these libraries will allow the program to perform faster linear algebra operations and remove the bottleneck of doing matrix operations on a CPU.

Graphics cards provide a greater advantage in parallelizable operations due to its architecture and purpose-built design for parallel computing. As it can be observed in Fig. 1, a graphics card is much unlike the architecture of a CPU with a few cores, system memory, and a cache. While modern CPUs are able to handle parallel computations to an extent, they are not purpose-built for parallel computing. A GPU consists of several Streaming Multiprocessors, (SMs) each with an array of embedded Streaming Processors (SPs) [5]. Two or more (architecture depending) SMs make up a building block and each multiprocessor’s SPs share the same instruction cache and control logic. The GPU also has a large buffer of high-bandwidth global memory for all of the blocks and every block also has its own block of shared memory [5]. Every thread has exclusive access to high-speed registers in the block for private memory storage regarding that specific thread. This specific architecture allows the GPU to excel at parallel computations. NVIDIA harnesses an architecture, first introduced in NVIDIA products, known as Single-Instruction, Multiple-Thread (SIMT) to handle parallel computing tasks [10]. SIMT

is an execution model where the multiprocessor manages the execution of hundreds of concurrent threads. The multiprocessor executes tasks in groups of 32 parallel threads, known as warps. Every warp is executing the same instruction at any given time. This structure of dividing a given task into warps and blocks allows SMs to be fully utilized on a graphics card for the purposes of fast parallel computing [10].

3 Methodology

In order to achieve the goal of implementing GPU support in a Gröbner basis algorithm, an algorithm was selected that utilized the F4-Matrix approach. The matrix approach is best suited for GPU integration since linear algebraic operations on matrices representing polynomial systems can benefit greatly from parallel processing techniques. Many of these computations are simple on their own, comparisons of constant values, scaling of rows and columns, and linear combinations of multiple polynomials. What makes these computations difficult to compute is not their complexity, but the sheer quantity of computations necessary to complete. This is what makes the use of a graphics card's many parallel cores advantageous over a CPU's fewer general-purpose cores; lower complexity and higher quantity of computations.

Various algorithms were explored such as F4, F5, F4/5, and F5C [8]. The F4/5 algorithm utilizes the matrix reduction style from the F4 algorithm and symbolic preprocessing from the F5 algorithm to create a new approach that takes full advantage of each version's strengths [9]. The authors of the F4/5 algorithm, Martin Albrecht and John Perry, included source code for a version that implemented the SageMath (Sage) system. Sage is an open-source mathematical system [10]. Thus, the F4/5 algorithm in Sage will be a benchmark for CPU-only performance to compare with the new implementation [9]. A representation of the algorithm in flowchart form can be seen in Fig. 2.

Before discussing how the algorithm computes the Gröbner basis, the ideal of a polynomial ring needs to be defined; as Gröbner bases are mainly defined within this context. A polynomial ring is formed from a set of polynomials that share a common variable and coefficients defined within the same space. For this algorithm, a subset of the ideal of a polynomial ring in n variables is considered if it satisfies three conditions: zero is contained in the subset, two polynomials are contained within the subset their addition will still be contained in the subset, and their multiple will also be contained within the subset. Under these conditions a Gröbner Basis can be defined in terms of the ideal.

The F4/5 algorithm has many parts which are broken down in Fig. 2 [9]. Each part may call upon the global list of polynomials, L , which may grow rapidly. A second global list of integers, G , identifies which indices of L will be in the Gröbner Basis. Additionally, the global variable, P , is a list of "critical pairs" of polynomials that may be used to create generated polynomials to add to L . All the input polynomials will be added to both L and G , and every possible combination

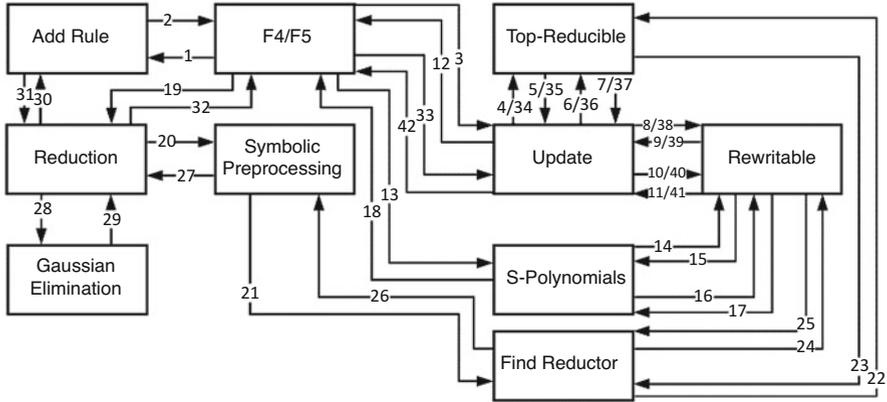


Fig. 2 F4/5 algorithm diagram [3]

of these polynomials will be sent through Update, which will return critical pairs that will be added to P . However, each pair must pass the F5 criteria, which is characterized through Top-Reducible, Rewritable, and Add Rule. Top-Reducible returns a Boolean that signifies if a monomial is reducible by performing long division of the parent polynomial by the minimal, if the result is zero it will return true. Rewritable determines if the signature of a monomial is a minimal signature, meaning that there does not exist in the system a signature that would make the current pairing redundant. If both conditions are met the critical pair will remain in the system and be given to Add Rule. A pair that fails any one of these checks will be dismissed. The ones that are added to P will be selected in groups by minimum degree to compute their S -polynomial.

The S -polynomial method will take the complete list of critical pairs, P , and construct a list of S -polynomials out of them. The S -polynomials are constructed by taking the polynomials associated with each critical pair and scaling each of them so that their leading terms cancel out if one is subtracted from the other. This will be done here only if the critical pair is still considered non-rewritable, as the check will also be performed one more time inside the S -polynomial portion. Afterwards the list of S -polynomials, S , will be passed to the Reduction method which first calls upon symbolic preprocessing and Gaussian elimination before any computations in Reduction. Symbolic preprocessing takes the list of S -polynomials, S , and determines which monomials and polynomial multiples could potentially be encountered while reducing the S -polynomials. It then sorts the resulting list of polynomials in order of decreasing signatures in order to avoid reductions to zero in the system and return the list, F . Lastly Gaussian elimination takes the system and constructs a representational matrix of the system and puts it into Reduced

Echelon Format (REF). After which reduction will produce the top-reduced set of the S -polynomials and return them to the main.

$$O(N) = 2^{D^N} \tag{1}$$

With that completing one loop of the F4/5 algorithm, the singular loop likely took little time to compute compared to the following loops. As the subset taken from the original system to be operated on is relatively small, with each additional loop of the algorithm, another polynomial is added to the subset and the process is repeated. Each method must perform additional actions during each run in writing data. This looping in on itself that the algorithm has to do leads to the very loose time complexity shown in Eq. (1). D representing the dimensional space of the polynomial system, and N representing the number of polynomials in the original system. After each polynomial has been processed from the original system using this process the algorithm will return the Gröbner basis of the system.

Since the authors of the F4/5 paper published a source code that can compute Gröbner bases in Sage, this was the obvious logical step to success [3, 11]. After getting the code working, the details of which will be explored in the next section, the algorithm was profiled to discover which functions in the execution took up the largest amount of time. Consistently, symbolic preprocessing and Gaussian elimination took up ~99% of the execution time with the former taking approximately one-third and the latter two-thirds of the execution time. Seeing how Gaussian elimination takes up two-thirds of the execution time, this indicated that there was great potential for a noticeable increase in performance. Gaussian elimination is an operation that can be parallelized with a GPU. Therefore, the primary design goal was to implement operation acceleration using CUDA.

4 Implementation

As stated before, the profiling of the F4/5 algorithm revealed that the Gaussian elimination steps on a CPU-only implementation in Sage takes up approximately two-thirds of the execution time when computing a Gröbner basis. This can be observed in Fig. 3, which showcases the runtime of a Katsura ideal which has been used in [3] as a benchmark with 8 variables in a finite field of size 32,003. Each row in Fig. 3 with a timestamp denoting elimination time is a function call to the Gaussian elimination method. This is where the GPU acceleration will be taking place upon implementation and negate the need for the CPU to do linear algebraic operations.

The source code for Albrecht and Perry's F4/5 algorithm was written in Python 2.7. The most recent version of Sage, version 9.0, utilizes Python 3 to execute scripts [11]. As expected, the source code was not initially compatible, however, Sage's source code allows for users to compile it for themselves and build the code with Python 2.7 support (Sage version 9.0 is the last to support this feature). It should be noted that the backbone of Sage runs off of a technology called Cython [12]. It was developed alongside (and for) Sage, but Cython and Sage are considered

```

Katsura
Multivariate Polynomial Ring in x0, x1, x2, x3, x4, x5, x6 over Finite Field of size 32083
CPU
=====
3 7 39 x 80, 39, 0 Gaussian Elimination Time(s): [0.029944]
4 12 99 x 155, 99, 0 Gaussian Elimination Time(s): [0.210109]
5 12 234 x 296, 234, 0 Gaussian Elimination Time(s): [1.094885]
6 7 386 x 369, 386, 0 Gaussian Elimination Time(s): [1.998583]
7 3 111 x 174, 111, 0 Gaussian Elimination Time(s): [0.275]
|L|: 74
L is GB: True
reductions to zero: 0
max. degree: 7
symbolic processing runtime: 2.724839
gaussian elimination runtime: 4.420122
total runtime: 7.22082
    
```

Fig. 3 Katsura-8 (32003) CPU-only Gaussian elimination timestamps

two separate technologies. Cython is a compiled version of Python and this is an important distinction since Python is an interpretive language. Since Sage is using Cython, using this platform allows for the use of external code libraries written in C/C++, which is exactly how CUDA will be integrated into Sage. Cython has a great deal of other features that pertain to interacting with definitions and functions in the C language [12]. However, this external library support was only half of the battle.

Getting a CUDA code to run externally in a static environment required for special compilation of the Gaussian elimination algorithm written for the GPU. To accomplish this, a public open-source repository on GitHub from Robert McGibbon was harnessed to bridge Sage and CUDA code libraries together [13]. His repository provides a template for getting CUDA code to compile into a static shared object library for use in Cython applications. Since Sage uses Cython, this allowed for Sage to talk directly to the graphics card. Figure 4 showcases a flowchart of how Sage will talk to the graphics card. When the F4/5 algorithm running in Sage needs a matrix row reduced, it will execute an external function in the shared object which will execute CUDA code and complete the necessary calculations.

The Gaussian elimination algorithm used in our GPU-accelerated implementation was based solely off the elimination algorithm in the F4/5 paper [3]. This elimination was relatively easy to implement in comparison to an algorithm like the Faugère–Lachartre algorithm, which is significantly more complex due to more execution logic [14]. The F4/5 Gaussian elimination is primarily a traditional row echelon form calculation with the exception of not swapping any rows or columns [3]. Figure 5 shows the source algorithm in the F4/5 paper. The input matrix is traversed row by row from the top downwards. Next, each column is traversed top to bottom and if the current term is a leading term, it will reduce all the rows below it. For reduction of rows below the leading term, the leading term’s row is first inverted, so the leading term in that row has a value of one. This makes scalar reductions much easier as it only takes a multiplication of negative one and the leading term row added to the target row to reduce its target entry in the same column as the leading term to zero. This process is repeated until the last row and then the algorithm terminates. Since the columns do not explicitly depend on one

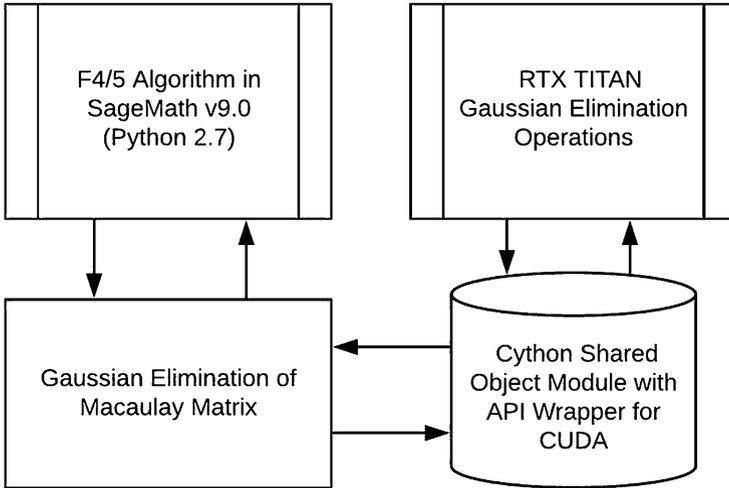


Fig. 4 Flowchart of Sage & CUDA communication

```

Input:  $F$  – a list of pairs  $(u, k)$  indicating that the product  $u \cdot \text{poly}(k)$  must be computed
Input:  $T$  – a list of all the monomials in  $F$ 
Result:  $\tilde{F}$  – a list of labelled polynomials
begin
   $m, n \leftarrow |F|, |T|;$ 
  denote each  $F_i$  by  $(u_i, k_i);$ 
  let  $A$  be the  $m \times n$  matrix such that  $a_{ij}$  is the coefficient of  $T_j$  in  $u_i \cdot \text{poly}(k_i);$ 
  for  $0 \leq c < n$  do
    for  $0 \leq r < m$  do
      if  $a_{rc} \neq 0$  then
        // Ensure that we are only reducing by leading terms
        if any  $a_{ri} \neq 0 \mid 0 \leq i < c$  then continue;
        ;
        rescale the row  $r$  such that the entry  $a_{rc}$  is 1;
        for  $r + 1 \leq i < m$  do // clear below
          if  $a_{ic} \neq 0$  then
            [ eliminate the entry  $a_{ic}$  using the row  $r;$ 
          ]
        break;
    ]
  ]
  let  $\tilde{F} = A \cdot T = \left[ \sum_{j=0}^{n-1} a_{ij} \cdot t_j \right]_{i=0}^{m-1};$ 
  return  $\tilde{F}$ 

```

Fig. 5 Gaussian elimination algorithm used in F4/5 [3]

another during row computations, the linear row operations on the matrix can be parallelized by the graphics card. After parsing the coefficient matrix back into a polynomial system list, the resultant matrix is passed back to the F4/5 algorithm in Sage [3]. This algorithm only gets called several times throughout the Gröbner basis computation but regardless it is a crucial step and great care was taken to ensure its validity in practice.

The input for the algorithm comes in the form of an ideal generated by a polynomial ring. These polynomial rings must be based on a specific number field. Some example number fields are real numbers, Real-Double-Float (RDF); a floating-point number in Sage, integer numbers, and finite fields [11]. As noted prior in this report, this project utilized finite fields in the testing of the GPU-accelerated implementation of F4/5. That decision was made due to the fact that the graphics card was unable to process other number fields properly due to inaccuracies with floating and double floating-point numbers. However, using a finite field meant that there is only a specific range of numbers that are valid in the field. This is implemented with integer modulus arithmetic. Field sizes 32,003 and 65,521 were tested as they are typically seen in other benchmarks and from the implementation of last year’s project. Custom CUDA kernel functions were written to accommodate integer modulus arithmetic and to do linear algebraic operations in a finite number field as well.

To execute the code, this project had direct remote access (via secure shell) to a high-end workstation at Cleveland State University that was sponsored by NASA GRC. The workstation is using one of NVIDIA’s most powerful GPUs, the TITAN RTX [15]. The TITAN RTX has 4608 CUDA Cores with a boost clock of 1770 MHz, 24 GB GDDR6 of video memory, and 576 Tensor Cores (unused for this project). The workstation’s CPU is a Xeon Silver 4116 clocked at 3.0 GHz with 12 cores and 24 threads. For system memory, the machine has 96 GB in total of Registered ECC server-grade memory. Note that even though these machine specifications provided our machine with a large pool of resources, Gröbner basis calculations are still lengthy and time-consuming based on the current algorithm’s architecture. Sage version 9.0 compiled with Python 2.7 support along with the proper drivers and dependencies were installed on this machine. Figure 6 displays the inside of the workstation residing at Cleveland State University.

Fig. 6 Workstation internals



The primary deliverables for this project were the source code, a currently running and valid installation on the workstation, and the creation of a code repository for the purposes of keeping this project open source. There is a public GitHub repository where all source code necessary to run this project is located. This project is licensed under the GNU General Public License v2.0 to ensure freedom and public access for all [16]. Sage version 9.0, CUDA drivers, software used to install Sage, and all source code will remain on the workstation until further notice.

5 Experimental Results

After successfully compiling the shared object with the Gaussian elimination algorithm, the F4/5 algorithm with GPU acceleration was ready to be tested. Several shell scripts and test benchmarks were set up to allow for multiple tests to be run successively one after the other. Many of the test executions with more variables took anywhere in the realm of a few hours to around a full day at least to compute. However, although some benchmarks did not finish, a consistent dataset was gathered to compile into an accurate result set. The first set of results to display is depicted in Table 1. This table showcases the Gaussian elimination statistics of a Cyclic ideal with 7 variables in a field size of 32,003. Cyclic ideals were used as a test case in Albrecht and Perry's F4/5 algorithm [9]. The overall percentage increases of CPU-only versus GPU-accelerated benchmarks can be observed in Figs. 7 and 8.

Figure 7 shows the increases for a finite field of size 65,221 and Fig. 8 shows the increases for a finite field of size 32,003. These numbers are strictly for the Gaussian elimination execution time to visualize how much faster the GPU is versus the CPU at the same Gaussian elimination algorithm.

Dealing with polynomial systems and Gröbner bases gives way for a unique and diverse dataset with multiple representations with which the data can be viewed. That is why three-dimensional plots were created to give a representation of the complexity of these benchmarks. The plots were created using a Python utility called Matplotlib. Figures 9 and 10 showcase a single benchmark's iteration statistics for a Katsura ideal with 8 variables in a finite field of size 32,003. Since each execution is split into iterations which take a polynomial system of some given minimal degree, it was a simple task to track the execution time for each iteration. The 3D axes are plotted as follows: iteration time, pairs of S-polynomials in the iteration, and minimal degree of the polynomial system in said iteration. Figures 9 and 10 showcase two different viewpoints of the same 3D plot to accurately depict the shape of the generated polygon. Figures 11, 12, and 13 showcase three viewpoints for the iteration statistics of a Cyclic ideal with seven variables in a finite field of size 32,003. In Figs. 9, 10, 11, 12, and 13, all green polygons represent the GPU results and the red polygons represent the CPU results.

As it can be seen in Figs. 9, 10, 11, 12, and 13, the GPU's results are consistently better than the CPU in all cases. The few exceptions are when the dataset is too small

Table 1 Cyclic-7 Gaussian elimination execution timestamps

Gaussian Elimination Benchmark - Cyclic 7 - Field Size 32,003			
S-Poly Pairs	CPU (s)	GPU (s)	CPU vs. GPU Increase (%)
1	0.0224	0.038	-70.00%
2	0.0483	0.0451	7.00%
6	0.2937	0.2377	19.00%
14	1.2089	0.9131	24.00%
34	4.8563	3.3268	31.00%
58	14.8531	8.8899	40.00%
100	36.0413	19.9679	45.00%
137	89.6144	44.9176	50.00%
199	135.7663	70.2492	48.00%
204	170.4749	81.8717	52.00%
200	216.9355	93.9579	57.00%
155	142.5655	69.4461	51.00%
92	209.563	92.789	56.00%
83	6.5576	4.5246	31.00%
131	11.886	8.3854	29.00%
157	7.5991	6.2813	17.00%
45	0.0035	0.0035	0.00%
68	0.0035	0.0035	0.00%
6	0.0006	0.0005	17.00%

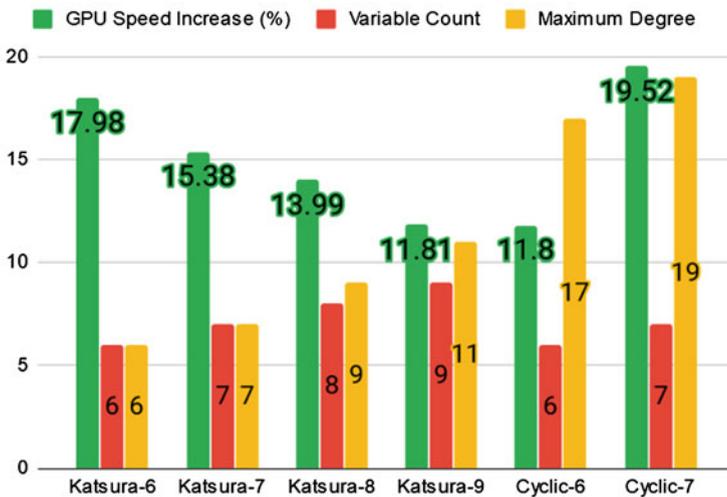


Fig. 7 CPU vs. GPU overall speed increase (65521)

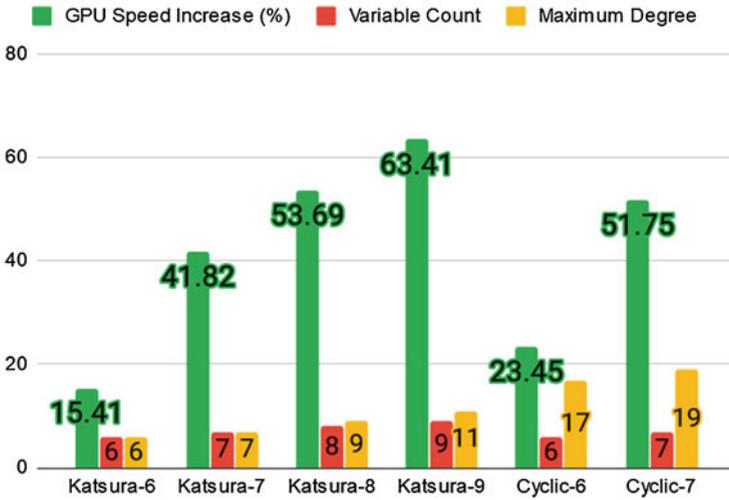


Fig. 8 CPU vs. GPU overall speed increase (32003)

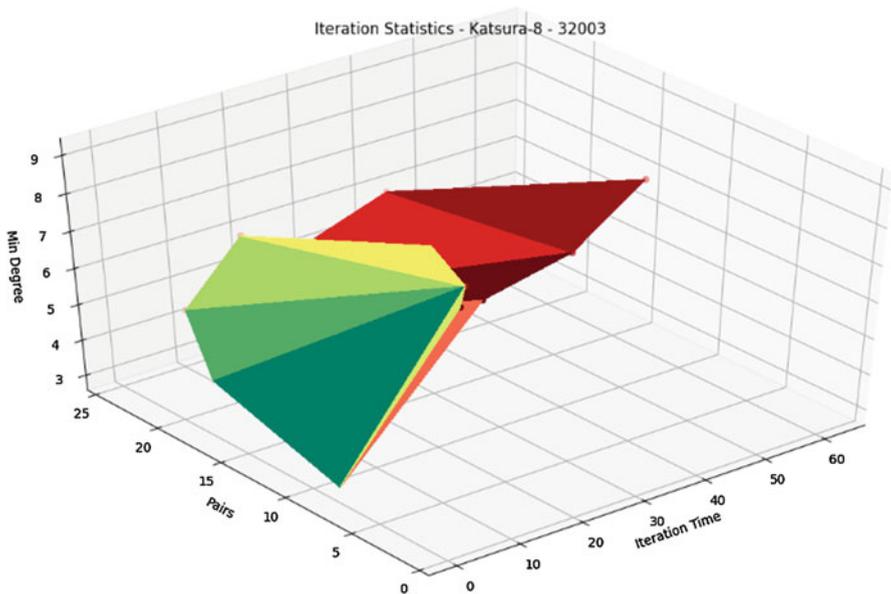


Fig. 9 Katsura-8 iteration statistics viewpoint #1 (65521)

to the point where the overhead of launching and copying data to the GPU is simply not faster than the CPU's total execution time. However, these outlier cases of small datasets have incredibly short runtimes as it is. Therefore, this project is finding the GPU to be a consistent performance increase to Gröbner basis calculations. In

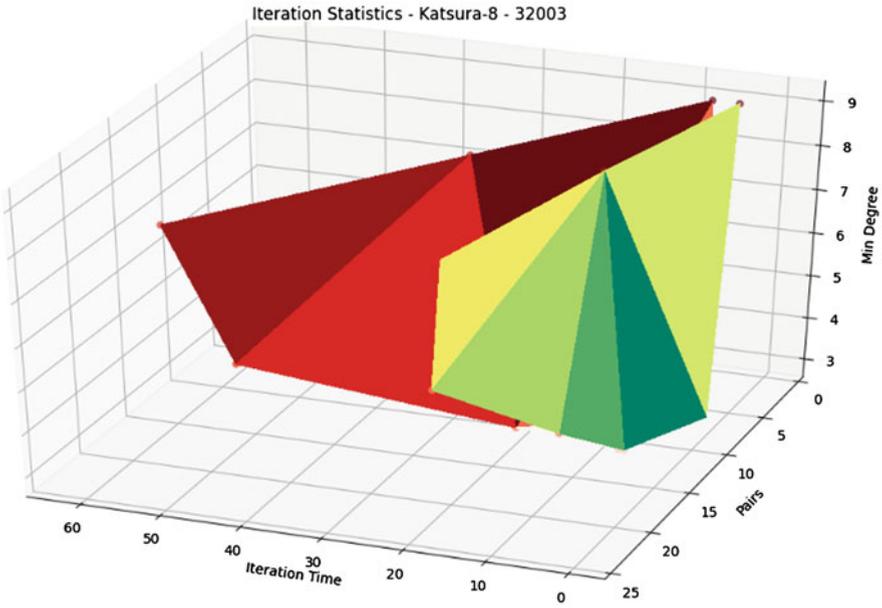


Fig. 10 Katsura-8 iteration statistics viewpoint #2 (65521)

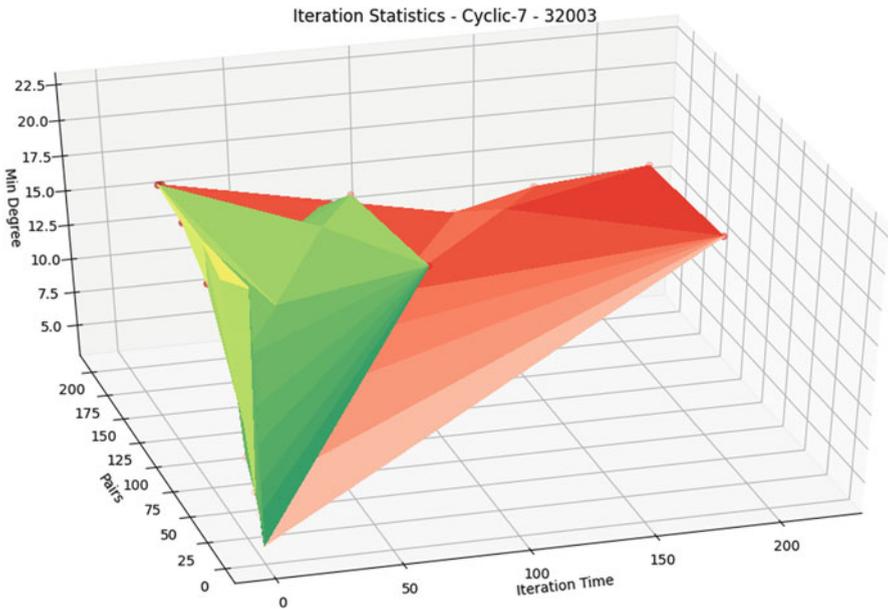


Fig. 11 Cyclic-7 iterations statistics viewpoint #2 (32003)

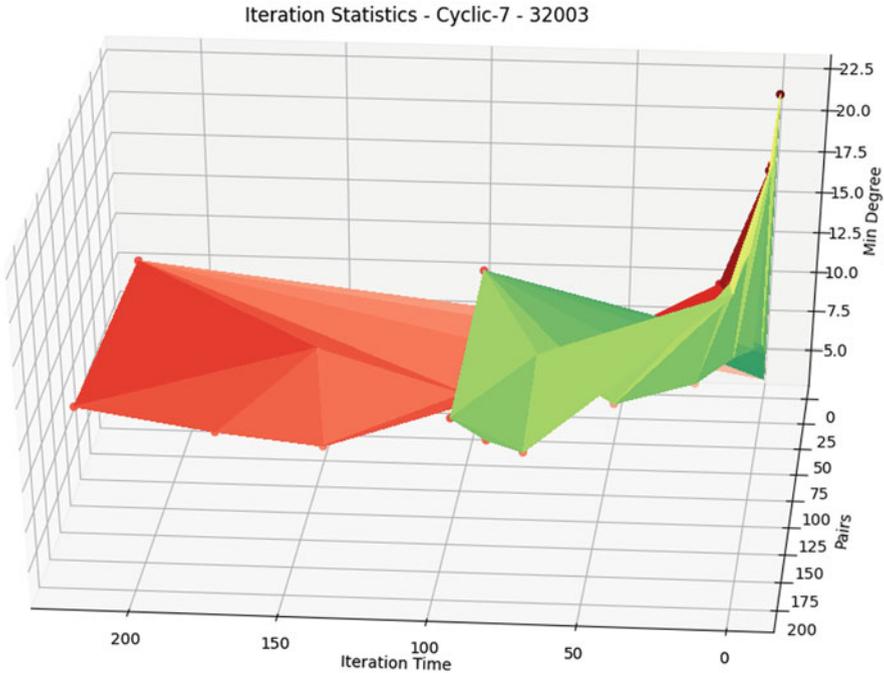


Fig. 12 Cyclic-7 iteration statistics viewpoint #1 (32003)

Fig. 8 concerning a finite field of size 65,521, the trend for performance decreases in Katsura ideals as the variables increase in count but performance increases for Cyclic ideals as the variable count increases. This could be due to a multitude of factors, chiefly pertaining to that specific field size or perhaps the performance increases become only marginal after increasing the dataset to a certain size. The 3D plots give insight into the complexity of Gröbner basis calculations in a way that 2D graphs cannot convey.

While there are observable increases in performance while using a GPU, there are still slower parts of the algorithm that need acceleration. This project’s focus was simply to accelerate the linear algebraic operations of Gröbner basis calculations. There are still other operations within the algorithm that could possibly benefit from GPU acceleration. Symbolic preprocessing and the conversion to and from a system of polynomials to a coefficient matrix representing that system are the primary culprits for the prolonged algorithm runtimes. Once the variable count for both Katsura and Cyclic ideals increases above 9 or 10, the benchmarks would execute for days at a time and only have completed a handful of iterations. It is unknown how long those benchmarks would have taken to finish executing. Overall, there is still much to be learned from the existing results and there is still a long way to go until these computations are operating at peak efficiency.

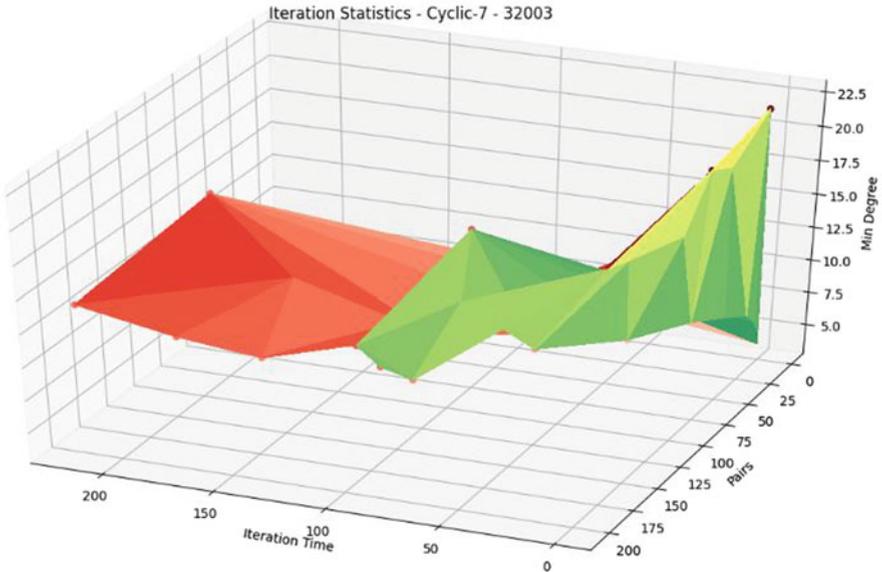


Fig. 13 Cyclic-7 iterations statistics viewpoint #2 (32003)

6 Conclusion

There is a long way to go until Gröbner basis calculations are fully optimized and operate at peak efficiency. Getting Sage to communicate with static C library functions and CUDA was a unique task that challenged the team to reach greater heights. However, with time and effort, this project succeeded in achieving the goal of implementing a Gröbner basis algorithm with GPU support. While it is not a perfect implementation, it is a step in the right direction toward optimizing these Gröbner basis algorithms even further than before.

NASA tasked this team with producing an algorithm that would utilize a GPU's processing ability to improve the computational efficiency of a Gröbner basis algorithm. Using Sage and the cuBLAS API from CUDA, the F4/5 algorithm with GPU acceleration was successfully implemented on the high-end workstation at Cleveland State University. The custom F4/5 algorithms introduces parallelization into an inefficient portion of the algorithm and provides notable improvement to the runtime of the algorithm. The benchmarks run on the workstation between CPU and GPU-accelerated runs of the F4/5 algorithm back up the previous statement, but also indicate that other functions, such as symbolic preprocessing and matrix parsing, itself could benefit from GPU implementations. Additionally, there are other Gröbner basis algorithms which may benefit more than F4/5 with GPU support. Utilizing a cluster of graphics cards on a single or a multiple machine setup could possibly be another method of further speeding up the computations of

Gröbner bases. However, special optimizations would have to be carefully seen through to make that solution a reality. These are just a couple ways this project could continue the improvement in Gröbner basis computational efficiency. Nonetheless, this project has proven that a Gröbner basis computation can be improved with parallelization on a GPU.

The public repository containing all source code and benchmark data for this project can be found at:

<https://github.com/markymark501998/Parallel-Grobner-Basis-GPU-Calculation>

Acknowledgments This project was sponsored by the NASA Glenn Research Center for the Senior Design course at the Cleveland State University.

References

1. J.-C. Faugère, A new efficient algorithm for computing Gröbner bases (F4). *J. Pure Appl. Algebra* **139**(1–3), 61–88 (1999)
2. J.C. Faugère, “A new efficient algorithm for computing Gröbner bases without reduction to zero (F5),” *Proceedings of the 2002 international symposium on Symbolic and algebraic computation - ISSAC 02*, 2002
3. A. Martin, P. John, “F4/5,” *arXiv.org*, 07-Oct-2010. [Online]. Available: <https://arxiv.org/abs/1006.4933>
4. C. Eder, J. Perry, F5C: A variant of Faugère’s F5 algorithm with reduced Gröbner bases. *J. Symb. Comput.* **45**(12), 1442–1458 (2010)
5. D.B. Kirk, W.-M.W. Hwu, *Programming massively parallel processors: A hands-on approach*, 3rd edn. (Morgan Kaufmann Publishers Inc, San Francisco, 2016)
6. A. Hylton, G. Henselman-Petrusek, J. Sang, R. Short, Tuning the performance of a computational persistent homology package. *J. Softw. Pract. Exp.* **49**(5), 885–905 (2019)
7. J. Sang, C. Lee, V. Rego, C. King, Experiences with implementing parallel discrete-event simulation on GPU. *J. Supercomput.* **75**(8), 4132–4149 (2019)
8. “cuBLAS,” NVIDIA developer documentation. [Online]. Available: <https://docs.nvidia.com/cuda/cublas/index.html>. Accessed: 20 Nov 2019
9. “cuSPARSE” CUDA toolkit documentation, [Docs.nvidia.com](https://docs.nvidia.com) (2019). [Online]. Available: <https://docs.nvidia.com/cuda/cusparses/index.html>. Accessed: 20 Nov 2019
10. “CUDA toolkit documentation v10.2.89,” CUDA toolkit documentation. [Online]. Available: <https://docs.nvidia.com/cuda/>. Accessed: 20 Nov 2019
11. “Sage,” SageMath mathematical software system. [Online]. Available: <https://www.sagemath.org/>
12. “Coding in Cython,” Coding in cython - sage developer’s guide v9.0. [Online]. Available: http://doc.sagemath.org/html/en/developer/coding_in_cython.html
13. R.T. McGibbon, “rmcgibbo/npcuda-example,” GitHub, 08-Sep-2019. [Online]. Available: <https://github.com/rmcgibbo/npcuda-example>
14. J. Faugère, S. Lachartre, “Faugère-Lachartre parallel gaussian elimination for gröbner bases computations over finite fields” (2019). Accessed 20 Nov 2019
15. “NVIDIA TITAN RTX is Here,” NVIDIA. [Online]. Available: <https://www.nvidia.com/en-us/deep-learning-ai/products/titan-rtx/>
16. “GNU general public license version 3 | open source initiative”, [Opensource.org](https://opensource.org/licenses/GPL-3.0) (2019). [Online]. Available: <https://opensource.org/licenses/GPL-3.0>. Accessed: 20 Nov 2019

Single Core vs. Parallel Software Algorithms on a Multi-core RISC Processor



Austin White and Michael Galloway

1 Introduction

Today, parallel processing is a common tool used for dividing tasks among several processors to relieve the burden from otherwise time-consuming and or computationally complex tasks. Parallel processing consists of breaking a process into multiple parts and having each part processed simultaneously [1]. It has uses in image processing, complex data modeling, resource intensive simulations, hosting many clients on a server, running several intensive programs simultaneously, and more. The potential applications of parallel processing increase as access to cloud computing resources become available to larger and larger workloads. Parallel processing has become an essential, ubiquitous tool for increasing performance on today's architectures and workloads.

This paper seeks to investigate the Raspberry Pi RISC, reduced instruction set computer [2], architecture's aptitude for parallel processing. The Raspberry Pi is a cheap, common tool used for software development [3]. The ability to utilize it for parallel processing allows it's modest but effective hardware to be utilized in an even greater number of standalone deployments [4]. We performed three experiments to analyze the Raspberry Pi's aptitude for parallel processing. We also explore OpenCV, Java, and Python as development libraries for parallel programs on the Raspberry Pi.

This paper is organized into an introduction, background, experiments, conclusion, future work, and references sections. The introduction section introduces the purpose of this paper. The background section explains key topics discussed in this paper. The experiments section includes the methodologies and results of the

A. White (✉) · M. Galloway
Western Kentucky University, Bowling Green, KY, USA
e-mail: austin.white682@topper.wku.edu; jeffrey.galloway@wku.edu

performed experiments. The conclusion section provides an analysis of the results and outcomes for each experiment. The future work section describes possible avenues future research could take.

2 Background

Parallel processing is conducted across multiple connected processors. Each processor executes asynchronously to the other processors on its designated independent workload. This is done by breaking the individual process into smaller pieces that each processor executes individually (Fig. 1).

Directly splitting a problem into parts that individual processors can execute is known as the divide and conquer algorithm. Typically processing is done sequentially, which is why processing in parallel using parallel processing techniques is multitudes faster [5]. One key concept in obtaining this speedup is minimizing the number interactions that process nodes must make with one another to compute their own workload. If nodes rely too much on communication between nodes, time will be wasted waiting for other nodes to perform their section of work, and any potential speedup may be reduced or rendered useless. Additionally, care should be taken to avoid race conditions [6] wherein the output of the code changes based on which node reaches a certain stage first. This includes when processors must modify

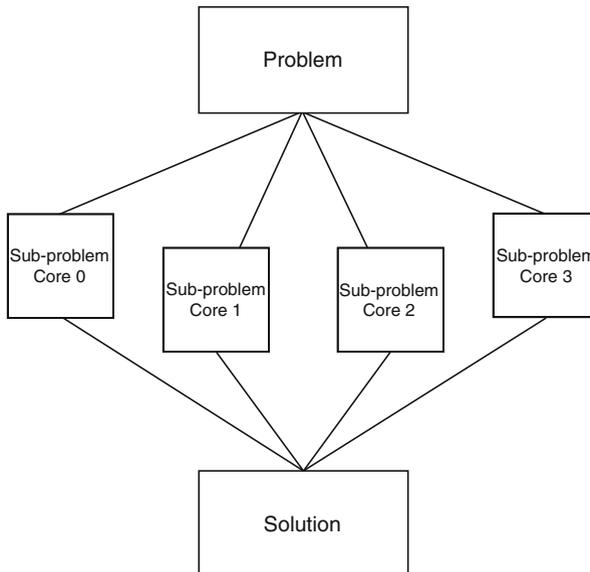


Fig. 1 This figure shows the idea of breaking up a problem into subproblems which can be run mostly/entirely independently

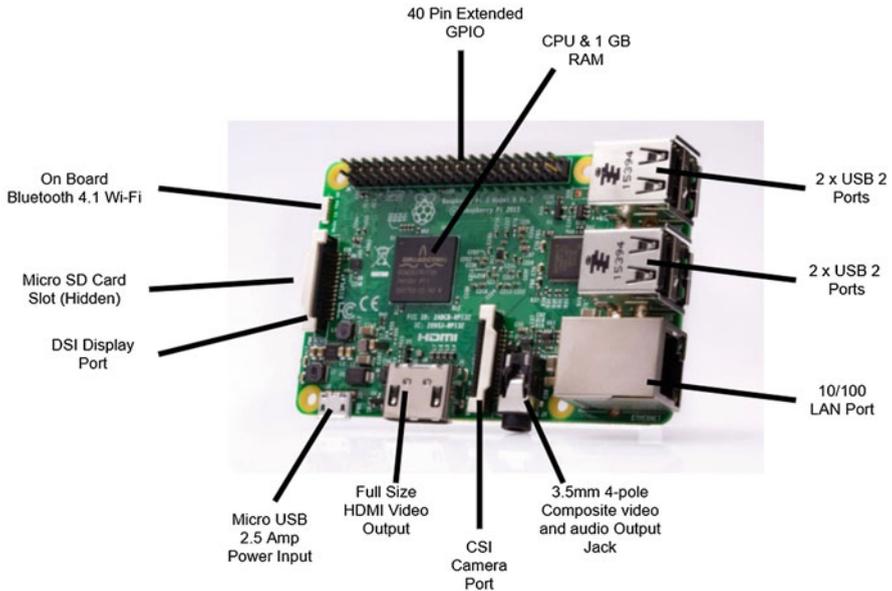


Fig. 2 The above diagram illustrates the hardware components of a Raspberry Pi 3 model B

or access the same data and when the results of their computations could depend on which node reached that point first. Here, we tried to avoid race conditions and too much communication by dividing processors among separate workloads with independent data, but some overhead still exists (Fig. 2).

Raspberry Pi is a low-cost, single-board microcomputer designed with the purpose of being used for education. They are a major aid in the area of computer science education. This is because the Raspberry Pi's architecture consists of a RISC CPU, GPU, multiple input and output circuits, general-purpose input/output pins, HDMI output, Wi-Fi adapter, Ethernet adapter, and SD/micro-SD card input on a single board. The Raspberry Pi sells for less than the price of 100 dollars [3] which makes it one of the most accessible and capable pieces of modern computer hardware. This paper's experiments were performed on the Raspberry Pi model 3B, which has a Quadcore ARM Cortex-A53, 64Bit processor. Because our Raspberry Pi is Quadcore, we have four available process nodes (cores) to distribute programs across, and we would expect that programs that can be efficiently run in parallel will be sped up by some constant multiple of the number of cores. Additionally, some additional overhead is expected due to communicating between processor cores. Because this communication overhead in our intended applications should not scale with the size of our data sets, we expect larger experiments to feel the effect of the constant multiple speedup even more.

Python is a commonly used programming language. It is open source and compatible with almost every OS. When installed, it contains a standard library

of packages that each provide their own functionality. Python Package Index is an online resource that contains a growing archive of thousands of packages that each add their own specific functionality to Python [7]. This includes packages that provide parallel processing functionality to Python.

C++ is a popular programming language that is based off of the older programming language C. For its users, it provides the same modularity as C but adds the implementation and functionality of an object-oriented programming language. C++ is a programming language that was created from the combination of these two design methods [8].

OpenCV is an open source library that contains various programming functions relating to video and image processing developed by the Intel Corporation. It was developed using C and C++ programming languages and is compatible with MAC OS, Windows, Linux, and other various operating systems. Many image and video processing software that utilize OpenCV are written in C, C++, Java, Python, and other programming languages. OpenCV was designed to be extremely lightweight and is commonly used on single board computer architectures, like the Raspberry Pi, for analyzing objects, security and intrusion systems, camera calibration, military applications and designs, medical image processing, satellite image processing, map applications, and more [9].

Bubble sort is an algorithm used for sorting elements in a single dimensional array. It works by going through the list of elements multiple times and each time, sequentially comparing adjacent elements and swapping them if necessary. It continues going through the list of elements in this manner, until it manages to go through the list without a single swap. At this point, the sorting is complete. At worst, it has a run-time efficiency of $O(n^2)$ [10] (Fig. 3).

MD5, Message-digest Algorithm 5, is a common hashing algorithm that we choose to use in our combinatorial password hashing program. We decided on this because of its availability within Java, fast hash time, and general acceptance as a mainstream hashing algorithm. It is the current mainstream information and network security tool for cryptography check and file check and is utilized in the databases of various sites for login password check [11].

Image and video processing are common parallel processes [12]. Image processing is a power- and time-consuming process that increases dramatically with the file size and number of pixels. Parallel-threaded GPUs are commonly used for high-quality image processing [13]. This is because CPUs have less computing cores and are nowhere near as capable at executing parallel processes. However, it is still possible to utilize CPUs for image processing.

Fig. 3 This image above is a Python implementation for the bubble sort sorting algorithm

```
def bubbleSort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n - 1 - i):
            if arr[j] > arr[j + 1]:
                temp = arr[j]
                arr[j] = arr[j + 1]
                arr[j + 1] = temp
    return arr
```

3 Motivation

Many corporations benefit from highly distributed computing. For example, Google maintains a database network that describes the Internet, which allows for efficient parallel searches of the web. Film and animation companies, such as Pixar, maintain GPU filled servers to edit, animate, and render video scenes with. Parallel processing benefits consumers directly too, by allowing them to more aptly handle lots of threaded workloads, such as web-browsing, spreadsheets, text editors, video streams, and programming IDEs, all at the same time. The demand for parallel computing is vast and growing with the need for compute-intensive applications, data-intensive applications, and network-intensive applications [14].

The results of this paper will analyze the robustness of Raspberry Pis for parallelizable workloads. Many high performance computing (HPC) applications are expensive, and thus there is a real argument to be made for the use of inexpensive individual computing units, such as Raspberry Pis which can make use of libraries such as MPI to distribute problems into a parallel domain [3]. Raspberry Pis have become a staple computing tool for small projects, because they are cheap and easy to use. Raspberry Pis use a RISC architecture, and examining the effectiveness of parallel algorithms on these computers will determine their potential for highly distributed computing in a cost-effective way. Also, RISC processors are designed to support high levels of pipe-lined, parallel execution, so the Raspberry Pi has architecture that suggests promising results [15]. The determination of a high potential will lead to a great increase of usage of the Raspberry Pi. The determination of a low potential will lead to a relatively small change in the usage of the Raspberry Pi. However, the data and analysis given in this paper will be beneficial to understanding the use cases of the Raspberry Pi, regardless of the degree of the determined potential.

Additionally, we used different languages to demonstrate the versatility and adaptability of the Raspberry Pi to different language environments. We used Python in our sorting tests, Java in our password hashing, and C++/OpenCV in our graphical rendering. Although there is no direct comparison to be made about the performance of these languages, the versatility of them to run parallel applications on the Raspberry Pi speaks to the generalized architecture of the Pi.

4 Experiments

4.1 *Parallel and Serial Sorting*

4.1.1 Methods

We implemented the bubble sort algorithm into Python. Using the multiprocessing library in Python, we created a pool of worker nodes and used the map function to split our unsorted array into sections for the workers to sort with the bubble sort

function. In theory, each of these worker nodes would divide the operations that go into bubble sort to create speedup in the sorting algorithm. However, many of the memory operations were blocking, and without a more specific articulation of how they were to divide the work, we did not observe this effect. We proceeded to run a series of tests on randomly generated arrays of varying sizes. Starting with arrays of size 2,000 to 10,000, we ran bubble sort in serial and parallel on 50 different arrays and computed the average time spent on each method. The Python program had an output of a CSV, comma-separated values, file that contains all of the generated data (Fig. 4).

4.1.2 Results

The following are graphical representations of the data achieved by following the previously described methodology. Timing was obtained with the `time.time()` function in Python (Figs. 5, 6, 7, and 8).

The figures above demonstrate that with this particular algorithm, we obtained no significant speedup from sorting in parallel. The blocking nature of the map function and the bubble sort algorithm make the parallelization more difficult and costly, reducing any potential gains of having multiple processors. However, some other sorting algorithms, such as merge sort, may benefit from this tactic, especially on larger data sets.

4.2 *Parallel and Serial Password Hashing*

4.2.1 Methods

Password hashing is the standard method of verifying user login information in modern cryptography. Password cracking is the process of using combinatorics and statistics to make educated guesses about which passwords may be contained in a database. Parallel processing threatens cybersecurity by reducing the amount of time it takes to try password hashes, although so-called “securely random” passwords remain safe from this attack. Trying all the permutations of a set of characters is called an exhaustive search, which, while rudimentary, proves to be a fitting benchmark of the ability of a processor to do arithmetic quickly. We tracked the computational cost of performing exhaustive search on words of various lengths using an alphabet of 64 characters in length. The implementation of our exhaustive search algorithm is in Java. We used the MD5 hash standard to encrypt our “passwords.” We used Java’s built-in threading capabilities to distribute one part of the total search space to each available node. Each node recursively builds a byte array that represents its section of the search space, and when it reaches the end of the word length that is trying, it computes the MD5 hash of the word. If it matches our password, it informs all the other nodes to end their search and

```

if __name__ == '__main__':
    printing = False
    f = open("data.csv","w")
    cnt = 10000

    rng = 1
    inc = 0
    tms = 50
    f.write("\n")
    for j in range(rng):
        f.write(str(cnt)+"\n")
        for k in range(tms):
            print(k);
            sze = 10000
            unsorted0 = [random.randint(1, sze) for x in range(cnt)]
            unsorted1 = unsorted0.copy()

            if printing:
                print("arr: ",unsorted0)
                print("arr2: ",unsorted1)

            #print("Ready. ")

            sstart = time.time()
            bubbleSort(unsorted0)
            send = time.time()
            f.write(","+str(send - sstart)+",")
            #print("Singular time in seconds: ", (send - sstart))

            pool = mp.Pool(mp.cpu_count())
            #print(mp.cpu_count())
            pstart = time.time()
            unsorted1 = [i for i in pool.map(bubbleSort, [unsorted1])]
            pend = time.time()
            f.write(str(pend - pstart)+"\n")
            pool.close()
            #print("Parallel time in seconds: ", (pend - pstart))

            if printing:
                print("arr: ",unsorted0)
                print("arr2: ",unsorted1)
        cnt += inc
        f.write(",,,=average(B"+str(3+j*(tms+2))+":B"+str(2+tms+j*(tms+2))+
        "),"+"=average(C"+str(3+j*(tms+2))+":C"+str(2+tms+j*(tms+2))+")"+"")+"\n")
        print("Cycle: "+str(j))
    f.close()

```

Fig. 4 The image above is the Python implementation of the code created and utilized to achieve our results

Fig. 5 The above table is the data achieved by following the described method of parallel bubble sort

Number of Elements	Time (s) Serial	Time (s) Parallel
2000	2.608108201	2.610664439
4000	10.69359688	10.67540613
6000	24.23157085	24.2098036
8000	43.36186999	43.35616867
10000	71.11271533	70.93550687

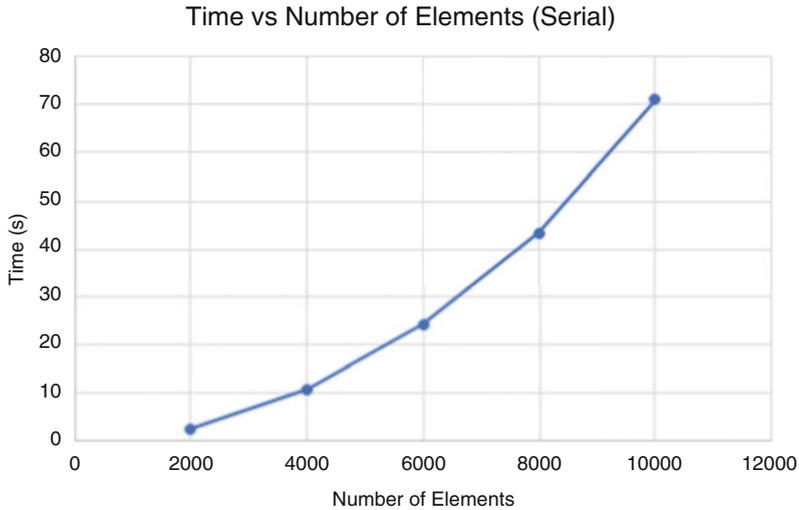


Fig. 6 This is a graph of how many seconds it took to run the sorting algorithm as a serial process while varying the number of elements in the sorted array

returns the found password and the time it took to find the correct combination. Otherwise, each node will exhaust their search space and the function will return null. We divided the search space by dividing the alpha into equal parts according to the number of nodes that we have available and setting the first character of our word to be from this subset of the alphabet. This way, each independent node equally divides the computational cost of searching every combination. Of course, to find the worst-case scenario we can simply input a password whose hash is known to not be in the search space, for instance, a word with characters not in the alphabet. In this way, we could time the total time it takes to search the space for a variety of lengths (Fig. 9).

The parallel implementation uses Java’s threading to do the division of the search, and the program terminates once all the threads have exhausted their search.

4.2.2 Results

After computing a complete search of the character space in serial and parallel, we recorded the time it took to complete in serial and the average time for each parallel

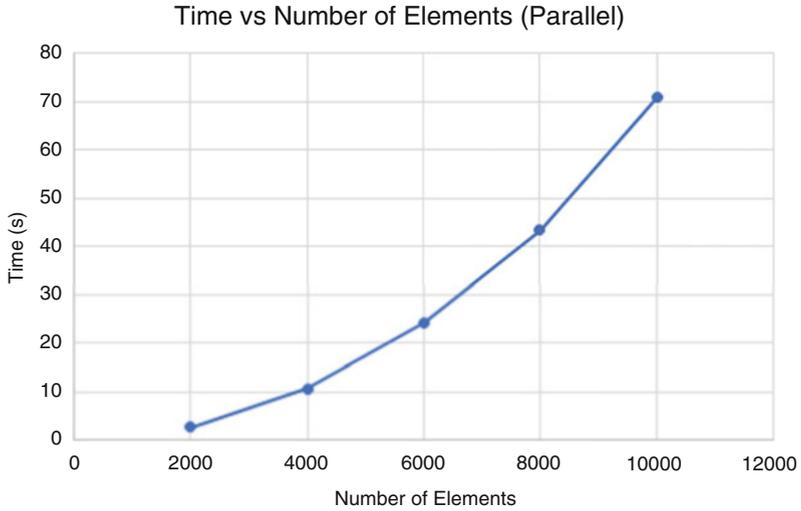


Fig. 7 This is a graph of how many seconds it took to run the sorting algorithm as a parallel process while varying the number of elements in the sorted array

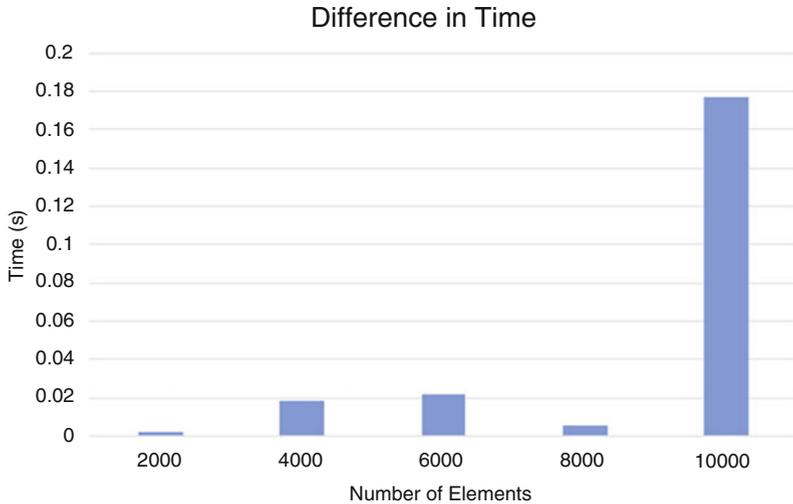


Fig. 8 This is a chart showing the difference in seconds between the serial and parallel processes for how long it took them to sort the arrays with a varying number of elements. The scale of the difference is within a margin of error, although the data sets were identical

```

import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Arrays;
/**
 * Serial exhaustive search for a password with known MD5 hash
 * @author Caedea #hitaker
 */
public class PasswordsParallel {

    public static void main(String[] args) {
        MessageDigest md;
        final String password = "*****";
        final byte[] passwordHash;
        try {
            md = MessageDigest.getInstance("MD5");
            passwordHash = md.digest(password.getBytes());

            long time = System.currentTimeMillis();
            Runnable operation = new Runnable() {
                @Override
                public void run() {
                    String name = Thread.currentThread().getName();
                    int t = Integer.parseInt(name.substring(name.length()-1));
                    byte[] newPH = Arrays.copyOf(passwordHash, passwordHash.length);
                    String s = paraSearch(newPH, password.length(), t);
                    System.out.println(name+": "+s+", "+(System.currentTimeMillis()-time));
                }
            };
            for(int j = 0; j<4; j++) {
                (new Thread(operation)).start();
            }
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
    }

    private static String paraSearch(byte[] h, int n, int t) {
        try {
            MessageDigest md2 = MessageDigest.getInstance("MD5");

            byte[] alpha = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789_".getBytes();
            byte[] hash = h;

            byte[] res = null;
            for(int i = 1; i<=n; i++) {
                for(int j = t*16; j<(t+1)*16; j++) {
                    res = new byte[i];
                    res[i-1] = alpha[j];
                    res = fill(res, hash, alpha, md2, i-1);
                    if(res!=null)
                        break;
                }
            }
            if(res==null)
                return null;
            return convert(res);
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        return null;
    }

    private static byte[] fill(byte[] word, byte[] hash, byte[] alpha, MessageDigest md2, int n) {
        if(n == 0) {
            if(same(md2.digest(word), hash))
                return word;
            return null;
        }
        byte[] res;
        for(byte a : alpha) {
            word[n-1] = a;
            res = fill(word, hash, alpha, md2, n-1);
            if(res!=null)
                return word;
        }
        return null;
    }

    private static boolean same(byte[] h1, byte[] h2) {
        for(int i = 0; i<h1.length; i++)
            if(h1[i]!=h2[i])
                return false;
        return true;
    }

    private static String convert(byte[] ba) {
        char[] ca = new char[ba.length];
        for(int i = 0; i<ba.length; i++)
            ca[i] = (char)ba[i];
        return new String(ca);
    }
}

```

Fig. 9 The image above is the Java implementation of the parallel search code created and utilized to achieve our results

Length of Password	Parallel Time (ms)	Serial Time (ms)
3	428	934
4	5504	15793
5	387556.5	951739

Fig. 10 Results from executing the brute force password hashing algorithm in parallel and in serial on the Raspberry Pi

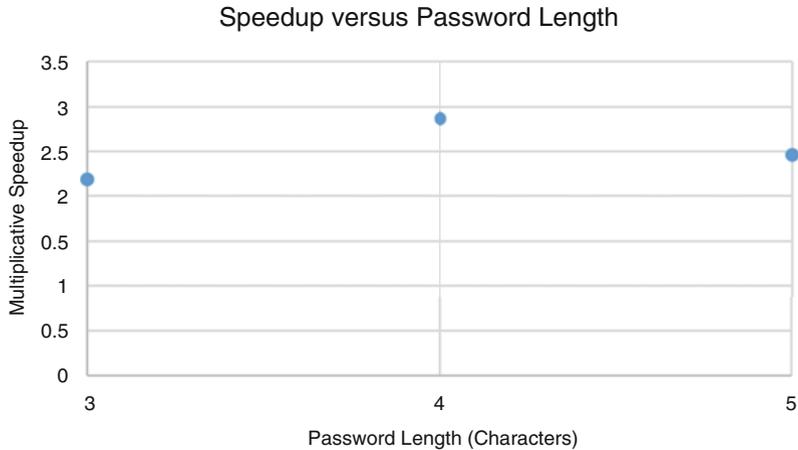


Fig. 11 The multiplicative speedup versus password length results from executing the brute force password hashing algorithm

thread. In this way, we could directly compare the amount of time spent checking for every possible hash with an input of a certain number of characters (Figs. 10 and 11).

After executing our code on the Pi, tests indicate a two to three times multiplicative speedup for searching the entire space on four independent threads as opposed to one. This is to be expected because there is no need for threads to block each other or communicate when searching independent spaces. This means the divide and conquer algorithm was highly effective in our implementation, and scaling the number of processors up by orders of magnitude will also increase our reach into the search space accordingly. Fortunately for security, the size of most passwords now is at least 8, which still takes an unobtainable amount of time on 64 characters for our implementation. But in principle, faster implementations on GPUs with many cores could brute force most smaller passwords. For this reason, the authors encourage everyone to use long securely random passwords.

4.3 *Parallel and Serial Graphics Rendering*

4.3.1 **Methods**

To execute the experiment, a virtual display was initialized. A virtual display was necessary because the goal was to test the processing speed at which the Raspberry Pi processed the frames of a video, serially and in parallel. The virtual display is the output of the processed image. The virtual display was created by executing the line “Xvfb :1 -screen 0 3840 × 2160 × 24 + 32”. This generates a virtual display that can support 2160p video quality using the Xvfb program. Afterward, the line “export DISPLAY=:1” was executed to set the virtual display to the active display. A video sample was procured, in varying levels of quality, and utilized as video rendering material. The same video sample was stored as a .mp4 file in 144p, 240p, 360p, 480p, 720p, and 1080p. The following pseudocode was executed on the Raspberry Pi to time the average speed it takes to render the frames in FPS for each quality of the video sample while executing the process serially, with two CPU cores in parallel and four CPU cores in parallel (Fig. 12).

The above pseudocode utilizes OpenCV libraries and operates by first opening the specified video sample .mp4 file. Next, it sets the number of CPU threads utilized to execute the process to 0, which OpenCV interprets as a command to process serially. It then starts a timer before processing the entire video sample frame by frame and exporting the frames to the virtual display. During this process, it is also counting the number of frames in the video sample. It then stops the timer and computes the total time taken in seconds for the process to execute. The next step is calculating the average FPS, frames per second, speed of the process and printing the value. Finally, it closes the video sample file. This same process is then repeated in parallel with the line “setNumThreads(getNumberOfCPUs())” telling OpenCV to execute the process with the maximum number of CPU cores present in the hardware. It is worth noting that getNumberOfCPUs() can be substituted with an integer value greater than 1 to execute the process with that many CPU cores. Also, rendering frames this way with the OpenCV libraries ignores the set FPS of a video, which allows the previous code to effectively act as a CPU graphics rendering process.

4.3.2 **Results**

One of the following graphs is a representation of the FPS percent difference (parallel FPS – serial FPS)/serial FPS, for each of the different tests. The percent difference for almost each was negative, indicating that the serial FPS was faster. The other graph displays the average FPS for each serial and parallel process executed for each video sample quality (Figs. 13 and 14).

The results displayed in the parallel processing FPS percent difference graph show that the serial process was always better than the parallel process. However, the

```
VideoCapture video0("<Video Quality>.mp4");
//Serial Processing
setNumThreads(0);

clock_t start0, end0;

start0 = clock();

int cnt0 = 0;

while(1){

    Mat frame;

    video0 >> frame;

    if(frame.empty())
        break;

    imshow("Frame", frame);
    waitKey(1);
    cnt0++;

}

end0 = clock();

double time_taken0 = ((double)(end0 - start0))/((double)CLOCKS_PER_SEC);

double testFps0 = (((double)(cnt0))/time_taken0);

cout << "<Video Quality> FPS Serial : " << fixed << testFps0 << setprecision(5);

cout << "\n";

video0.release();

////////////////////////////////////

VideoCapture video1("<Video Quality>.mp4");
//Parallel Processing
setNumThreads(getNumberOfCPUs());

clock_t start1, end1;

start1 = clock();

int cnt1 = 0;

while(1){

    Mat frame;

    video1 >> frame;

    if(frame.empty())
        break;

    imshow("Frame", frame);
    waitKey(1);
    cnt1++;

}

end1 = clock();

double time_taken1 = ((double)(end1 - start1))/((double)(CLOCKS_PER_SEC));

double testFps1 = (((double)(cnt1))/time_taken1);

cout << "<Video Quality> FPS Parallel : " << fixed << testFps1 << setprecision(5);

cout << "\n";

video1.release();
```

Fig. 12 This is the pseudocode of the C++ program utilized to process the videos frame by frame at the Raspberry Pi's maximum speed

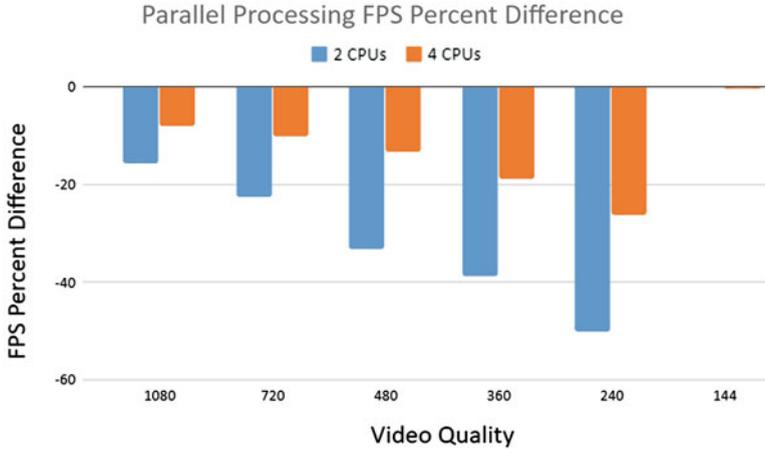


Fig. 13 This is a chart of the percent difference in FPS (frames per second) in the results achieved by the parallel process from the serial process

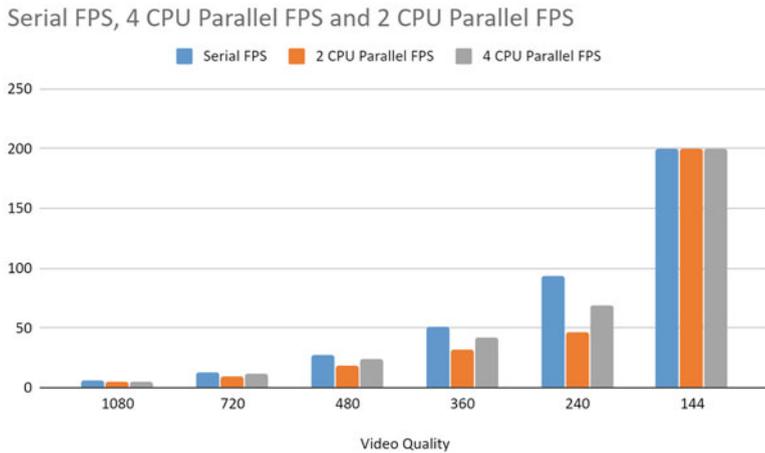


Fig. 14 This is a chart of the FPS (frames per second) results achieved by the serial process and the parallel processes

difference between the two decreased as the quality of the video sample increased. Also, the 4 CPU parallel process was twice as close to the performance of the serial process than the 2 CPU parallel process.

The results displayed in the serial FPS, 4 CPU parallel FPS, and 2 CPU parallel FPS graph show that the FPS generally decreases as the quality of the video sample increases. Also, overall the performance from best to worst was the serial FPS, 2 CPU parallel FPS, and 4 CPU parallel FPS.

5 Conclusion and Future Work

5.1 *Parallel and Serial Sorting*

While the Raspberry Pi is capable executing on 4 cores, we saw no difference between the performance of the serial or parallel methods, within the standard error. This is in direct contrast to the findings from Sujatha et al. [10]. This led us to believe that our implementation of the parallel computing algorithm may not be processing as a parallel process correctly. This could be due to inefficiencies in utilizing the parallel processing map function in Python with our bubble sort implementation. We also believe that the bubble sort algorithm is not well suited to parallelism because it contains blocking as processes work on the same sections of the array.

A better future implementation of this experiment may attempt to use a language with a lower-level implementation of control over parallel processes. In future work, making use of multi-threading in C could allow for different results when applied to a sorting algorithm. The results of our first experiment draw no conclusions as to the pros or cons of parallel sorting, as the Python parallel mapping function may not be efficiently distributing the load of the array. Using pthreads in C could allow us to specifically designate which parallel threads we want to create and distribute this to different cores on the Raspberry PI. In this way, we could at least ensure the array was distributed to the cores efficiently. Additionally, the merge sort algorithm is built around the divide and conquer paradigm. Using merge sort could potentially distribute sections of the array to sort on different cores.

5.2 *Parallel and Serial Password Hashing*

Using parallel nodes and the divide and conquer paradigm to partition the search space multiplicatively improved the performance of our exhaustive search algorithm. There was overall sufficiently little overhead in the parallelism to allow for a speedup that had a real affect on the ability to complete certain hash brute force attacks. The degree of the speedup was not the same for the 3-, 4-, and 5-character length passwords, but the degree of speedup did not generally increase or decrease as the character length increased either.

Further research could use random entry passwords to compute the average time it takes to find a given password in the search space. This would be useful to better analyze the real-world performance of such a setup, since the vast majority of passwords will not be found in the worst-case scenario timings. Additionally, the majority of passwords are not indeed random. So-called rainbow tables contain a list of commonly used real-world passwords and their respective hashes. In practice, much password cracking makes use of combinations of these passwords and other English words with well-known substitution operations to test many possible password combinations against a known password hash. These operations do benefit from parallel speedup, such as the well-known password cracking software Hashcat, which can make use of GPU units as well.

5.3 Parallel and Serial Graphics Rendering

Results show that for the graphics rendering process that required extremely little CPU computational power, in this case the 144p quality video, processing it serially or in parallel yields almost the same results. However, for processes immediately above this in required CPU computational power, processing serially is faster than processing in parallel. Although, as processes became more CPU computationally intensive, parallel processing became closer to serial processing in terms of processing performance. We believe that this was due to the overhead of executing the process in parallel. This also suggests that at a certain level of computational intensity, the video rendering process would execute faster in parallel than serially. However, due to the physical components, most likely the weaker than typical PC RISC CPU, of the Pi, this occurs much later than in a typical multi-core CPU. Also, it is worth noting that executing the process with 4 CPU cores halved the FPS percent difference, in comparison to executing the process with 2 CPU cores. Overall, the results show that for the complex task of graphics rendering, the serial form of the process produced a faster average execution time than the parallel form.

Further research could include increasing the quality of the video sample further and finding the point at which the parallel process exceeds the serial process in speed, as suggested by our results. Also, experimentation with how the process is parallelized could be pursued. An example of this would be rendering one frame on each core simultaneously, instead of splitting up the task of rendering one frame across the cores.

5.4 Conclusion

Overall, the Pi was demonstrated to be capable of parallel processing in several languages, although the benefits of this depended heavily on the algorithms. For a task that requires a lot of interconnected computations such as bubble sorting, the parallel form of the process was no different than the serial. For a more simply parallelized task such as password cracking, the parallel form of the process was always faster than the serial on the Raspberry Pi, and the degree of execution time increase varied with the intensity of the process. For instance, for a more complicated task such as graphics rendering, the parallel form of the process was slower than the serial on the Raspberry Pi, but the parallel process appeared to catch up to the serial process as the intensity of the process increased. These results show that the Raspberry Pi is highly capable at executing specific parallel processes faster than their serial form. Therefore, we conclude that the implementation of independent multi-threading processes for execution on Raspberry Pis should be encouraged because the architecture shows potential for utilization with specific types of parallel algorithms.

Acknowledgments We would like to thank Caeden Whitaker for contributing heavily to this paper and experimentation.

References

1. L. Hangye, S. Xuelian, Z. Jingchun, Study on architecture of photogrammetric parallel processing system based on cluster computing, in *2009 International Conference on Environmental Science and Information Application Technology*, vol. 1 (2009), pp. 378–381
2. J. Hennessey, D. Patterson, *Computer Architecture* (Elsevier LTD, Oxford, 2017)
3. N.S. Yamanoor, S. Yamanoor, High quality, low cost education with the Raspberry Pi, in *2017 IEEE Global Humanitarian Technology Conference (GHTC)* (2017), pp. 1–5
4. D.V. Diwedi, S.J. Sharma, Development of a low cost cluster computer using Raspberry Pi, in *2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN)* (2018), pp. 11–15
5. J. Liu, Y. Wu, J. Marsaglia, Making learning parallel processing interesting, in *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops PhD Forum* (2012), pp. 1307–1310
6. H.Y. Chen, Race condition and concurrency safety of multithreaded object-oriented programming in java, in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 6 (2002), 6pp
7. A. Kumar, S.P. Panda, A survey: how Python pitches in it-world, in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)* (2019), pp. 248–251
8. Y. He, The course choice between c language and c++ language, in *2009 4th International Conference on Computer Science Education* (2009), pp. 1588–1590
9. S. Solak, E. Doğru Bolat, Real time industrial application of single board computer based color detection system, in *2013 8th International Conference on Electrical and Electronics Engineering (ELECO)*, (2013), pp. 353–357
10. K. Sujatha, P.V.N. Rao, A.A. Rao, V.G. Sastry, V. Praneeta, R.K. Bharat, Multicore parallel processing concepts for effective sorting and searching, in *2015 International Conference on Signal Processing and Communication Engineering Systems* (IEEE, 2015)
11. Z. Yong-Xia, Z. Ge, Md5 research, in *2010 Second International Conference on Multimedia and Information Technology*, vol. 2 (2010), pp. 271–273
12. Y.S. Gener, A. Yildiz, S. Goren, Low-cost and low-power video filtering with parallel many cores, in *2015 9th International Conference on Electrical and Electronics Engineering (ELECO)* (IEEE, 2015)
13. A. Asaduzzaman, A. Martinez, A. Sepehri, A time-efficient image processing algorithm for multicore/manycore parallel computing, in *SoutheastCon 2015* (2015), pp. 1–5
14. Y. Li, Z. Zhang, Parallel computing: review and perspective, in *2018 5th International Conference on Information Science and Control Engineering (ICISCE)* (2018), pp. 365–369
15. S. Lindsay, B. Preiss, On the Performance of a Multi-Threaded RISC Architecture, In proceedings of Canadian conference on electrical and computer engineering (IEEE). (Date Added to IEEE Xplore: 06 August 2002); IEEE Xplore, <https://doi.org/10.1109/CCECE.1993.332333>. 14-17 Sept. 1993

MPI Communication Performance in a Heterogeneous Environment with Raspberry Pi



Oscar C. Valderrama Riveros and Fernando G. Tinetti

1 Introduction

MPI (message passing interface) is the standard communication library for distributed memory parallel computers [15]. The MPI focus is the message passing programming model in a *computer cluster*; that is, a group of computers interconnected by a computer interconnection network, mostly in a LAN (Local Area Network) environment. Current multicore processors [2, 10, 11] are easily harnessed together for parallel computing with MPI, given they are used in standard PCs, which also include networking hardware. Besides, SBC such as the Raspberry Pi is also used for parallel computing in several scenarios. The Raspberry Pi has been considered for parallel computing from different points of view:

- Low-cost computing platform: it has been introduced/proposed as a complete and functional computer and, thus, it is possible to construct a computing cluster with similar functionalities to that of a cluster of PCs to a fraction of cost and performance. From this point of view, it is highly recommended at least as a low-cost parallel computer, where distributed memory parallel algorithms can be developed and tested [1, 8]. And some of them have been specifically constructed for teaching parallel computing [12].

O. C. Valderrama Riveros

III-LIDI, Facultad de Informática, UNLP, La Plata, Argentina

Universidad Cooperativa de Colombia, Ibagué, Colombia

F. G. Tinetti (✉)

III-LIDI, Facultad de Informática, UNLP, La Plata, Argentina

CIC, Prov. de Buenos Aires, Buenos Aires, Argentina

e-mail: fernando@info.unlp.edu.ar

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_33

451

- Low-energy consumption (parallel) computing: the simple hardware design and implementation lead to a very-low power consumption compared to that of a standard PC or HPC (high-performance computing) node of computation. From this point of view, there have been several works showing a successful Mflop (millions of floating point operations) per Watt or, at least, a better Mflop per Watt than standard computing platforms [7].

Given the high availability of standard computing clusters for parallel computing, as well as the newly available Raspberry Pi clusters, it is likely that both parallel computing platforms will be combined for several parallel computing tasks. Performance optimization and analysis on these heterogeneous computing platforms would be enhanced taking into account several well-known issues such as workload balance and communication performance. Furthermore, the communication/computing ratio given the processor or cluster node computing power (e.g., Mflop/s, millions of floating point operation per second).

We have conducted a set of experiments for analyzing the communication performance with MPI, so that every possible communication overhead is taken into account, measured, and reported. We have selected the simplest experiment: a single PC and a single Raspberry Pi, so that the experimentation would provide the best-case performance and, thus, it could be considered as a granularity threshold.

We have considered the last three Raspberry Pi models at the time of this writing: 3B, 3B+, and 4. The Raspberry Pi 3 B is equipped with a 100 Mb/s Ethernet communication hardware. With the launch of the Raspberry Pi 3 B+, the communication hardware was upgraded to Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbp/s). Finally, the Raspberry Pi 4 includes a native Gigabit Ethernet. Thus, the experiments would provide the direct communication performance for parallel applications with MPI between Raspberry Pi 3 B, Raspberry Pi 3 B+, and Raspberry Pi 4. In order to avoid any communication interference/noise, we have used a direct (cable) interconnection between the PC and the Raspberry without any other traffic in the network. Besides, the PC has a Gigabit Ethernet interface so that if there is some hardware performance, penalty would be due to the Raspberry Pi hardware (and, eventually, software) interface. Two simple evaluation benchmarks were used: one of them uses the most straightforward MPI operations `MPI_send()` and `MPI_Recv()`, while the other one uses MPI-2 passive one-sided communication [3].

2 Related Work

The Raspberry Pi has been proposed and used in many contexts of parallel and distributed computing. The HPC area in particular has taken advantage of the Raspberry Pi low cost and support of a full-fledged operating system so that parallel computing is straightforward by using an implementation of the MPI

specification. A detailed guide to build a raspberry-based cluster in order to create a low-cost supercomputing is presented in [20] and a low-cost cluster computing platform is presented in [8], and there are a large number of similar papers. In terms of custom applications of clusters, studies referred herein [17, 18] present implementations of continuous monitoring in health environments. The works [5, 9] use parallel/cluster computing for DNA (deoxyribonucleic acid) analysis and cyber security, respectively.

The parallel and distributed computing academic environment is extensively using the Raspberry Pi hardware and hardware configurations. Several examples of using Raspberry Pi for teaching specific aspects of parallel and distributed computing can be mentioned:

- Heterogeneous cluster application in [25].
- IoT (Internet of Things) E-learning [26].
- Introducing or learning the first steps in supercomputing and parallel computing learning [13, 14, 24].

In terms of the performance analysis, the Raspberry Pi has been used in several different ways, such as for building a cluster for image processing [21], and as a testbed for distributed computing in wireless mesh network experimentation [19]. However, the Raspberry Pi has limitations beyond those related to (CPU) raw processing performance. More specifically [4, 16], explain that communication bandwidth limits imply, in turn, some limits to the Raspberry Pi cluster computing performance. A proposal to use a USB 3.0 Gigabit adapter is introduced in [4, 16].

Our work documents the problem of combining (fundamentally different) heterogeneous computing, which, in turn, introduces specific changes in traditional cluster management binaries and libraries. Besides, we experimentally show the different communication performance of the currently most powerful Raspberry Pi models with benchmarks using MPI interprocesses communication operations.

3 Heterogeneity Issues

One of the first problems found in a Raspberry Pi-PC heterogeneous environment is that of the binary differences. Unlike standard homogeneous clusters or even PC heterogeneous environments (usually based on x86 or x64 Intel architectures) is that each platform will be constrained to use its own binary code. Different PCs could use a shared file system folder/directory to store common and compatible binary code, and many Raspberry Pi could have a similar setting, but the program binary used in a PC architecture cannot be used in a Raspberry Pi. This heterogeneity starts at the operating system level, where

- PC would use some Linux version such as that of Ubuntu [6] or CentOS [23], depending on institution local policies of manageability and the number of cluster nodes.

- Raspberry Pi would use its own Linux distribution, which could be based on Ubuntu, such as Raspbian, but specifically adapted/developed for the Raspberry Pi architecture [22].

The standard operating system services and configuration, such as network, and secure shell are made in the standard way on each platform (PC and Raspberry Pi) for MPI operation.

As with the case of the operating system, the MPI applications have to be individually installed in each platform. More specifically in this work, both communication benchmarks were compiled and linked as usual with MPI implementation tools. We have chosen to have the same binary names and binary locations relative to the user home user directory/folder. Take into account that code/application versioning cannot be implemented by a single shared directory as in a homogeneous/compatible nodes cluster.

4 Hardware and Benchmarks Details

The hardware used in our experiments has been as simple as possible in order to have few but representative best-case scenario in each case. We have used a single PC with 8GB RAM, Gb/s network interface, and Linux Ubuntu 18.04 LTS operating system. We have experimented with three currently available Raspberry Pi models: Raspberry Pi 3 B, Raspberry Pi 3 B+, and Raspberry Pi 4. Actually, those models can be considered the best Raspberry Pi models from the point of view of computation and communication performance, that is, the ones most likely used in current Raspberry Pi installations for HPC. We do not describe the hardware configuration of each Raspberry Pi model because it is already documented in the manufacturer's site, and we have used them without any hardware change. The same operating system was installed in all Raspberry Pi: Raspbian Buster (2020-02-13-raspbian-buster), the current operating system version available from the Raspberry Pi manufacturer. The PC Ethernet interface is directly connected by an Ethernet cable to the Raspberry Pi Ethernet interface.

We have used two simple communication performance benchmarks, as mentioned before, which we will call:

- SendRecvPP (Send-Receive Ping-Pong), which uses the most straightforward MPI operations `MPI_send()` and `MPI_Recv()`. A single one-way data transfer is computed as half the time of two actual data communications: an initial send-recv from the "Ping" process to the "Pong" process immediately followed by a send-recv from the "Ping" process to the "Pong" process, as an echo data communication.
- OneSidedPP (One-Sided Ping-Pong), which uses MPI-2 passive one-sided communications. The benchmark uses the combination of three MPI operations: `MPI_Win_lock()`, `MPI_Put()`, and `MPI_Win_unlock()` for synchronizing the

actual data send operation from the “Ping” process to the data receive operation at the “Pong” process.

Each SendRecvPP single experiment will transfer twice the data (from the Ping process to the Pong process and in the reverse direction) and compute the one-way data communication as half the time of both data communications. The OneSidedPP would also transfer data twice, but in each case with the same operation: MPI_Put(). Moreover, each OneSidedPP communication needs process synchronization (with its corresponding overhead time) in order to make sure that the data have been actually transferred from the Ping process to the Pong process and vice-versa.

We have experimented with several message lengths, and for each length we have used a number of rounds, providing the average time as the experiment result. The range of lengths is rather broad: from a single byte to about 260 MB. We have found that 260 MB is on the maximum memory data amount that can be handled in the MPI Raspberry Pi environment.

5 Performance Results

Using a single benchmark, a direct comparison of Raspberry Pi models timing communication performance can be made. For example, Fig. 1 shows the OneSidedPP results for relatively small length messages. The Raspberry Pi 4 is the best one, as expected, and the specific values quantitatively show the communication performance gain of newer Raspberry Pi models. Besides, the graph also shows that for the three Raspberry Pi models the initial latency or, directly, the message latency for message lengths of less than 1024 bytes is about constant, in approximated values: 0.001s for the Raspberry Pi 3, 0.0006 for the Raspberry Pi 3B+, and 0.0004s for the Raspberry Pi 4.

The SendRecvPP benchmark has almost the same performance characteristics as those described with the OneSidedPP one: (a) newer Raspberry Pi models are better than previous ones, with almost the same relative better values as those shown in Fig. 1, and (b) the latency value about constant for messages lengths up to 1024 bytes.

Figure 2 shows that the SendRecvPP benchmark provides better performance values than the OneSidedPP benchmark for message lengths up to 1024 bytes in the Raspberry Pi 4. Actually, the SendRecvPP benchmark provides almost between 50% and 70% better latency performance values than the OneSidedPP benchmark in all three Raspberry Pi models, for message lengths up to 1024 bytes. The performance difference is basically due to the usage of the MPI process synchronization operations needed to ensure the correct data arrival time at the destination process when the MPI one-sided operations are used.

For message lengths between 1024 bytes (1 KB) and 260 KB, both benchmarks have almost the same communication performance in all three Raspberry Pi models. For very large messages, that is, message lengths greater than 260 KB, the one-sided

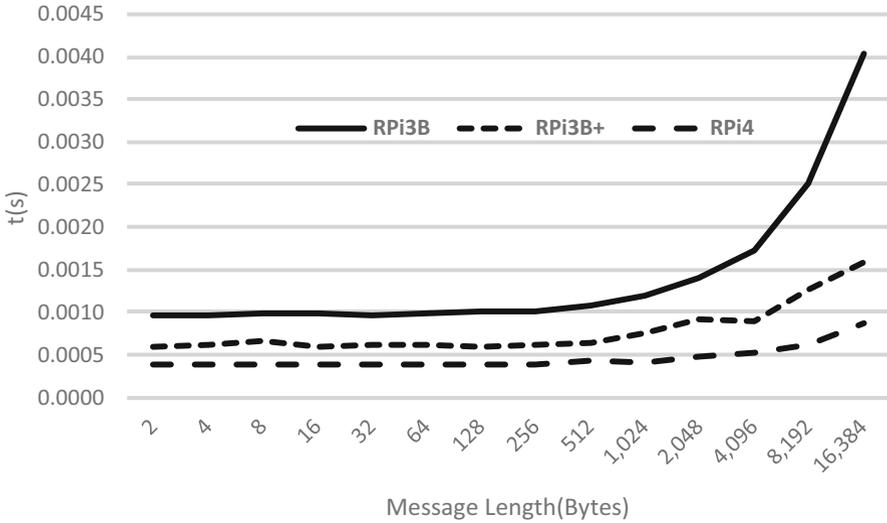


Fig. 1 OneSidedPP benchmark results, relatively small messages

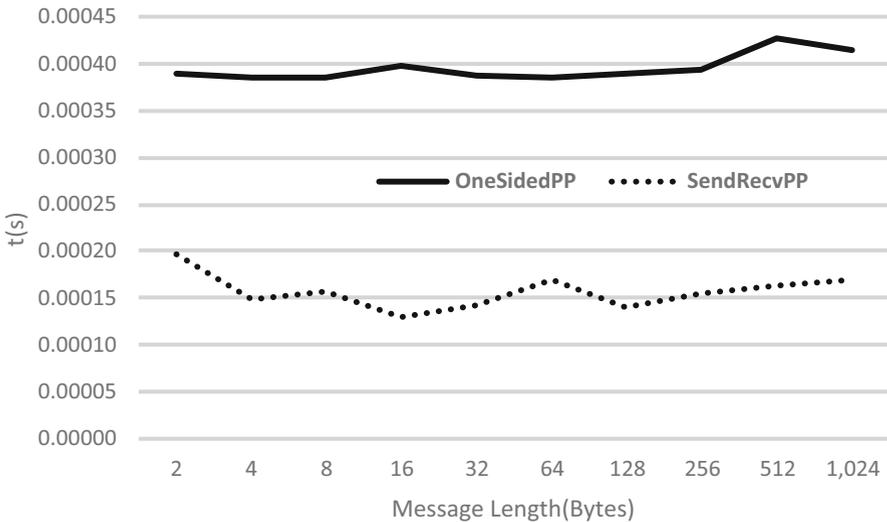


Fig. 2 SendRecvPP and OneSidedPP benchmark results, small messages, Raspberry Pi 4

data communications perform between 20% and 40% better than the simple Send-Recv operations in the newest Raspberry Pi 4.

Figure 3 shows a summary of performance results in terms of data bandwidth obtained with both benchmarks in the three Raspberry Pi models for message lengths between 30 KB and 300 MB:

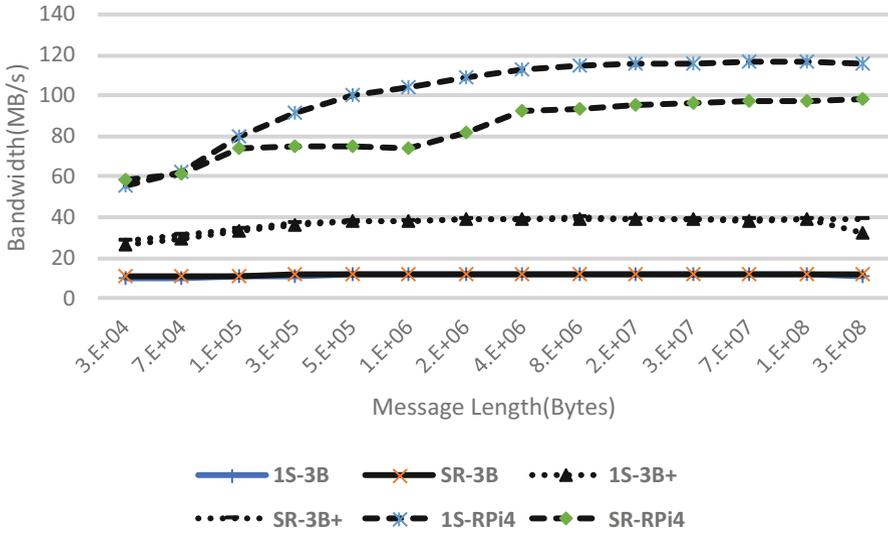


Fig. 3 SendRecvPP and OneSidedPP bandwidth results

- Results obtained in the Raspberry Pi 3B model are shown in solid lines (1S-3B for OneSidedPP and SR-3B for SendRecvPP): both benchmarks provide similar results, between 9 MB/s and 11 Mb/s.
- Results obtained in the Raspberry Pi 3B+ model are shown in dotted lines (1S-3B+ for OneSidedPP and SR-3B+ for SendRecvPP): both benchmarks provide similar results, better than those in the Raspberry Pi 3B, between 30 MB/s and 40 MB/s.

Results obtained in the Raspberry Pi 4 model are shown in dashed lines (1S-RPi4 for OneSidedPP and SR-RPi4 for SendRecvPP): both benchmarks provide better results than in older Raspberry Pi models, and the OneSidedPP benchmark (labeled as 1S-RPi4) obtains better results than the SendRecvPP benchmark (labelled as SR-RPi4), as noted before. Almost the best bandwidth is obtained by the OneSidedPP benchmark in the Ethernet GB/s network: 120 MB/s.

6 Conclusions and Further Work

We have been able to evaluate the MPI communication performance in heterogeneous environments combining PC and different Raspberry Pi newer models (Raspberry Pi 3B, 3B+, and 4). Installation and configuration of the entire system is straightforward in terms of individual basic tasks in each platform. The heterogeneity implies to be careful and carry out extra tasks for generating and maintaining each binary and MPI parallel program version, because it is no longer

possible to use a single shared file system for storing (parallel program/processes) binaries.

We have been able to evaluate two performance communication benchmarks: one based on standard Send-Receive MPI communications operations and the other based on one-sided MPI communication operations. Both benchmarks provide the same communication performance regarding to the following:

- Each newer Raspberry Pi model outperforms the previous one, that is, the Raspberry Pi 4 is the one with the best communication performance and the Raspberry Pi 3B is the worst one among the three evaluated models.
- Communication latency for relatively small size messages (e.g., messages less than 2 KB long) is almost constant. The specific communication latency time is found to be worse when using one-sided MPI communications operations.
- For relatively large messages (e.g., messages greater than 2 KB long), communication performance is almost the same in each individual Raspberry Pi model independently of the specific MPI communication operations used (one-sided vs Send-Recv).
- The one-sided MPI communication operations outperforms the Send-Recv ones only in the Raspberry Pi 4, and for messages greater than 300 KB long.

With previous performance values and performance behavior, we expect to define and experiment with granularity measurements and experimentation. More specifically, we expect to find specific values for identifying some useful value/s of the computing/communication relationship so that:

- Discard parallel computing in case the expected parallel performance would not be *acceptable*. Conceptually, computing time in each platform should be greater than the communication time needed for sending/receiving data on those platforms. Computing time would be strongly dependent on the application and parallel programming/algorithm to be used.
- At least provide a rough a priori performance analysis and evaluation to be taken as reference depending on the implemented algorithm.

References

1. P. Abrahamsson, S. Helmer, N. Phaphoom, L. Nocolodi, N. Preda, L. Miori, M. Angriman, J. Rikkilä, X. Wang, K. Hamily, S. Bugoloni, Affordable and energy-efficient cloud computing clusters: The Bolzano Raspberry Pi cloud cluster experiment, in *Proc. of the IEEE International Conference on Cloud Computing Technology and Science* (2013)
2. Advanced Micro Devices, Inc., Retrieved March 2020, <https://www.amd.com/en/ryzen>
3. B. Barrett, G. Shipman, A. Lumsdaine, “Analysis of implementation options for MPI-2 one-sided,” Recent advances in parallel virtual machine and message passing interface: 14th European PVM/MPI User’s Group Meeting, Paris, France (2007)
4. G. Bernabé, R. Hernández, M.E. Acacio, Parallel implementations of the 3D fast wavelet transform on a Raspberry Pi 2 cluster. *J. Supercomput.* **74**, 1765–1778 (2018)
5. O.O. Buyuk, A.Y. Camurcu, A novel actual time cyber security approach to smart grids, 2018 6th International Istanbul Smart Grids and Cities Congress and Fair (ICSG), Istanbul, 2018

6. Canonical Ltd. Ubuntu, "The leading operating system for PCs, IoT devices, servers and the cloud | Ubuntu", Retrieved March 2020, <https://ubuntu.com/>
7. M.F Cloutier, C. Paradis, V.M Weaver, "Design and analysis of a 32-bit embedded high-performance cluster optimized for energy and performance," Co-HPC '14: Proceedings of the 1st International Workshop on Hardware-Software Co-design for high performance computing (2014)
8. D.V. Diwedi, S.J. Sharma, Development of a low cost cluster computer using Raspberry Pi, 2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN), Lonavala, India, 2018
9. F. Habibie, Afiahayati, G.B. Herwanto, S. Hartati, A.Z. Kusuma Frisky, A parallel ClustalW algorithm on multi-Raspberry Pis for multiple sequence alignment, 2018 1st International Conference on Bioinformatics, Biotechnology, and Biomedical Engineering - Bioinformatics and Biomedical Engineering, Yogyakarta (2018)
10. Intel Corporation, Intel core processor family, Retrieved March 2020, <https://www.intel.com/content/www/us/en/products/processors/core.html>
11. Intel Corporation, Intel Xeon Processors, Retrieved March 2020, <https://www.intel.com/content/www/us/en/products/processors/xeon.html>
12. B. Levandowski, D. Perouli, D. Brylow, Using embedded Xinu and the Raspberry Pi 3 to teach parallel computing in assembly programming, 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Rio de Janeiro, Brazil (2019)
13. C. Madritsch, T. Klinger, A. Pester, Work in progress: CoCo.Cluster of computers in remote laboratories, in the challenges of the digital transformation in education. International Conference on Interactive Collaborative Learning (ICL 2018), part of Advances in Intelligent Systems and Computing, ed. by M. Auer, T. Tsiatsos, Springer, New York. 917, 569–576 (2018)
14. S.J. Matthews, J.C. Adams, R.A. Brown, E. Shoop, Portable parallel computing with the Raspberry Pi, Proc. of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)
15. Message passing interface forum, MPI: A message-passing interface standard, version 3.1 (2015)
16. L. Miori, J. Sanin, S. Helmer, A platform for edge computing based on Raspberry Pi Clusters, in *Data Analytics. BICOD 2017. Lecture Notes in Computer Science*, ed. by A. Cali, P. Wood, N. Martin, A. Poulouvassilis, vol. 10365, Springer Nature, New york
17. S. Misbahuddin, A.R. Al-Ahdal, M.A. Malik, Low-cost MPI cluster based distributed inward patients monitoring system, 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA), Aqaba (2018)
18. S. Misbahuddin, M.M. Ibrahim, A.M. Alnajjar, B.Q. Alolabi, A.F. Ammar, Automatic Patients' Vital Sign Monitoring by Single Board Computer (SBC) Based MPI Cluster, 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia (2019)
19. T. Oda, D. Elmazi, T. Ishitaki, A. Barolli, K. Matsuo, L. Barolli, Experimental results of a Raspberry Pi based WMN testbed for multiple flows and distributed concurrent processing, 2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), Krakow, Poland
20. A. Pajankar, Raspberry pi supercomputing and scientific programming, Apress (2017)
21. R.F. Rahmat, T. Saputra, A. Hizriadi, T.Z. Lini, M.K.M. Nasution, Performance test of parallel image processing using open MPI on raspberry PI cluster board, 2019 3rd international conference on electrical, telecommunication and computer engineering (ELTICOM), Medan, Indonesia (2019)
22. Raspberry Pi Foundation, Download Raspbian for Raspberry Pi, Retrieved March 2020, <https://www.raspberrypi.org/downloads/raspbian/>
23. The CentOS project, CentOS project, Retrieved March 2020, <https://centos.org/>
24. P. Turton, T.F. Turton, PiBrain - a cost-effective supercomputer for educational use, 5th Brunei International Conference on Engineering and Technology (BICET 2014), Bandar Seri Begawan (2014)

25. J. Wolfer, A Heterogeneous Supercomputer Model for High-Performance Parallel Computing Pedagogy, 2015 IEEE Global Engineering Education Conference (EDUCON), Tallinn (2015)
26. M. Yamada, T. Oda, K. Matsuo and L. Barolli, Design of an IoT-based E-learning testbed, 2016 30th international conference on advanced information networking and applications workshops (WAINA), Crans-Montana (2016)

A FPGA-Based Heterogeneous Implementation of NTRUEncrypt



Hexuan Yu, Chaoyu Zhang, and Hai Jiang

1 Introduction

The tendency of quantum computing has been proved to imperil the majority of classical cryptography schemes. Under these circumstances, the post-quantum cryptography schemes that are resistant to quantum computing have attracted people's notice, which leads to the demand for innovations. Lattice-based cryptography is a promising quantum-safe cryptography family [1], both in terms of fundamental security properties and applicability to both traditional and emerging security problems such as digital signature, encryption/decryption, key exchange, and fully homomorphic encryption (FHE). NTRU (Nth-degree truncated polynomial ring unit) cryptosystem has been fully accepted into the IEEE P1363 standards as part of the specifications for lattice-based public-key cryptography (IEEE P1363.1) since 2009.

However, compared to classic number-theoretical cryptosystems, the computational complexity of NTRU is still relatively high, which may impede their large-scale application. The traditional implementations of NTRUEncrypt, as well as other cryptosystems, concentrate upon conventional processors such as CPUs, whereas, there were some fundamental problems which prevent those cryptosystems from increasing operations per watt including: (1) Memory wall. The incompatible frequency between memory and processor has become a bottleneck. It may take several clock cycles between initiating a request and retrieving data. (2) Data dependency between operations and software/hardware constraint prevent parallelization.

In the era of heterogeneous computing, the special-purpose computing device can be accessed by the CPU to offload some computations to achieve lower cost

H. Yu (✉) · C. Zhang · H. Jiang

Department of Computer Science, Arkansas State University, Jonesboro, AR, USA

e-mail: hexuan.yu@smail.astate.edu; chaoyu.zhang@smail.astate.edu; hjiang@astate.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_34

461

or higher power efficiency. Some former GPU versions [2, 3] have clarified that NTRUEncrypt can benefit a lot from parallelization accelerators. FPGA (field-programmable gate array) has been widely explored as a hardware accelerator due to its reconfigurability and fast turnaround time, and it is one of the most cost-effective devices where tasks can be parallelized in significant measure.

This work explores the inherent parallelism of the NTRU-Encrypt cryptosystem on an FPGA-based heterogeneous system (CPU+FPGA) using OpenCL (open computing language) to maximize the throughput of NTRUEncrypt, considering the FPGA resource constraints such as on-chip memory, logic block, registers, and memory bandwidth. OpenCL is a parallel and cross-platform processing standard from the Khronos Group. Utilizing OpenCL on an FPGA can reduce the time-cost comparing with hardware description language (HDL) development and offer significantly higher performance than the one on other hardware devices such as CPUs, GPUs, and DSPs at much lower power consumption.

The rest of this paper is organized as follows.

Section 2 briefly describes the mathematics background and relevant procedures of NTRUEncrypt, including key generation and encryption/decryption. Section 3 introduces the framework of OpenCL-based NTRUEncrypt cryptosystem and our programming model. The implementation and results analysis, including the comparison between FPGA and GPU implementations, are given in Sects. 4, 5, respectively. The results show that NTRUEncrypt is computationally inexpensive within FPGA. The conclusion is drawn in Sect. 6.

2 NTRUEncrypt

In the past few decades, lattice-based cryptography has become one of the most intriguing areas of mathematical cryptography. The NTRUEncrypt is based on the shortest vector problem of lattice theory [4].

2.1 Notation

NTRU operations are based on objects in a truncated polynomial ring $R = \mathbb{Z}[X]/(X^N - 1)$ with convolution multiplication, and all polynomials in the ring have integer coefficients and degree at most $N - 1$ with $X^N \equiv 1$:

$$a = a_0 + a_1X + a_2X^2 + \dots + a_{N-2}X^{N-2} + a_{N-1}X^{N-1} \quad (1)$$

The product

$$C(X) = a(X) * b(X) \quad (2)$$

is given by

$$C_k = a_0b_k + a_1b_{k-1} + \dots + a_{N-1}b_{k+1} = \sum_{i+j=k \bmod m} a_i b_j \quad (3)$$

In particular, if we write $a(X)$, $b(X)$, and $c(X)$ as vectors

$$a = [a_0, a_1, \dots, a_{N-1}], b = [b_0, b_1, \dots, b_{N-1}], c = [c_0, c_1, \dots, c_{N-1}] \quad (4)$$

then $c = a * b$ is the convolution product of two vectors having a size of N positions.

The NTRU algorithm is defined by the following parameters:

- N . The degree parameter, defining the degree $N - 1$ of the polynomials in R .
- q . A large modulo. Polynomial coefficients are reduced modulo q .
- p . A small modulo. The coefficients of the message are reduced modulo p in decryption.
- \mathcal{L}_f . Private key space, fixing the polynomial form to define the number of positive ones for the private key f . The negative ones are fixed by $d_f - 1$.
- \mathcal{L}_g . Public key space, fixing the polynomial form to define the number of positive and negative ones for the random polynomial g used to calculate the public key.
- \mathcal{L}_r . Blinding value space, fixing the polynomial form to define the number of positive and negative ones of the random polynomial r used in the encryption process.
- \mathcal{L}_m . Plaintext space. NTRUEncrypt requires the message to be in a polynomial form, therefore the need for d_m to define the form of the message to be encrypted.

NTRU is specified by three public integer parameters (N, p, q) which represent the maximal degree $N - 1$ for all polynomials in the truncated ring R , a small modulus, and a large modulus, respectively, where it is assumed that N is prime, q is always larger than p , and p and q are co-prime. Note that p and q need not be prime, and four sets of polynomials \mathcal{L}_f , \mathcal{L}_g , \mathcal{L}_r , and \mathcal{L}_m (a polynomial part of the private key, a polynomial for the generation of the public key, the message, and a blinding value, respectively) are all of the degree at most $N - 1$ [5].

2.2 Key Generation

The key generation includes the generation of the private key (f, f_p) and the public key h [6]. Choose random polynomials f and g from R with small coefficients, typically $\{-1, 0, 1\}$ for $p = 3$. Then compute f_p , i.e., the inverse of $f \bmod p$ defined by

$$f * f_p = 1 \bmod p \quad (5)$$

Bob creates a public key h by choosing elements $f, g \in R$, computing the mod q inverse f_q^{-1} of f , and setting

$$h \equiv f_q^{-1} * g \pmod{p} \quad (6)$$

Bob's private key is the element of f . Bob also precomputes and stores the mod p inverse f_p^{-1} of f .

2.3 Encryption

In order to encrypt a plaintext message $m \in R$ using the public key h , Alice selects a random element $r \in R$ and forms the ciphertext

$$e = r * h + m \pmod{q} \quad (7)$$

This ciphertext hides Alice's messages and can be sent safely to Bob.

2.4 Decryption

In order to decrypt the ciphertext e using the private key f , Bob first computes

$$a \equiv f * e \pmod{q} \quad (8)$$

He chooses $a \in R$ to satisfy this congruence and to lie in a certain prespecified subset R_a of R . He next does the mod p calculation

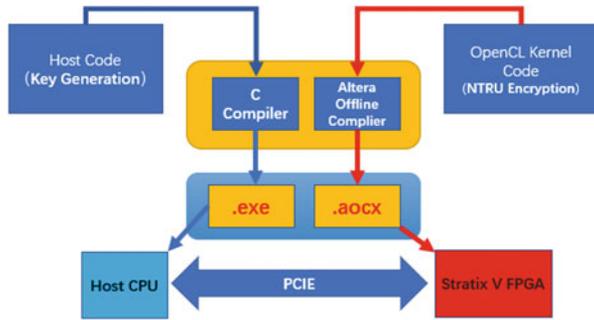
$$f_p^{-1} * a \pmod{p} \quad (9)$$

and the value he computes is equal to m modulo p .

3 A Heterogeneous NTRUEncrypt Platform

The OpenCL standard is an open programming model for accelerating algorithms on a heterogeneous computing system. OpenCL extends the C-based programming language for developing portable codes on different platforms such as CPU, GPU, DSP, and FPGA. In this section, we describe our CPU-FPGA heterogeneous framework of NTRUEncrypt Cryptosystem. As shown in Fig. 1, our OpenCL system

Fig. 1 FPGA Implementation of NTRU using OpenCL



implementation is divided into two components: the host PC and FPGA device. More details will be described as follows.

3.1 Overview

Where FPGAs shine in aspects of energy efficiency is at configurable logic and fixed precision computations. In lattice-based cryptography, it is exactly this property that makes FPGAs advantageous. The aim of this design is to mitigate the efficiency problem by providing an FPGA-based implementation that can run NTRUEncrypt cryptosystem on parallel hardware accelerators.

As shown in Fig. 1, the NTRUEncrypt Cryptosystem written in OpenCL consists of two parts: a host program for initialization and management and kernels that define the compute-intensive tasks. The host is responsible for device memory management, transferring data to devices, queuing work for devices, and error management. The host CPU communicates with the Stratix V FPGA via a PCI-E interface. The host can then get and utilize the encryption results, again via the PCI-E interface. OpenCL simplifies the transformation methodology of turning logic into FPGAs. As we have seen in Sect. 2, the operations consisted in this cryptosystem primitives are modular addition and multiplication operations; there contain neither logarithmic nor exponentiation operations. The polynomial multiplication and addition of NTRU encryption correspond to XOR and AND operations, respectively. Hence, they are easier to implement. However, this advantage is accompanied by the cost of intensive computation and storage requirements which may impede the adoption of NTRU over number-theoretic cryptosystems. In this case, the optimization of massive XOR and AND operations is of vital importance to a 256-bit level NTRU encryption.

3.2 *Programming Model*

OpenCL supports three programming models: data-parallel programming model, task-parallel programming model, and the hybrid of the two. Generally, NTRU encryption can be executed in both data-parallel and task-parallel programming models. This is because, firstly, the size of its input/output per loop is fixed and decided by chosen parameters and all the input plaintext will be used by the same encryption logic, which allows NTRU encryption to be executed in data-parallel form. Secondly, the XOR and AND operations toward its single bit are independent; thus it is also capable of being executed in a task-parallel fashion during each cycle.

Compared to CPU and GPU which execute the kernel on different cores, FPGA offers advantages by transforming the kernel into dedicated and pipelined hardware circuits. Data-parallel portions of the NTRUEncrypt algorithm are executed on FPGA as kernels, which are OpenCL functions.

Our design aims to minimize the number of resources, thus achieving higher energy efficiency while sustaining the same throughput. How much an algorithm uses the FPGA is an important aspect of FPGA programming. An algorithm only utilizes the logic it requires within the FPGA, leaving the remaining logic unused and consuming minimal power. Therefore, large power savings can be achieved by designing a kernel to meet only the needs of the target system, something that is not possible on CPU and GPU technologies.

In this heterogeneous system, the host launches encryption kernels across a 2D grid of work-items to be processed by the FPGA. Conceptually, work-items can be thought of as individual processing threads that each execute the same kernel function. Work-items have a unique index within the grid and typically compute different portions of the result. Work-items are grouped into work-groups, which are expected to execute independently from one another. The parallelism can be achieved since the kernel can be duplicated inside FPGA using pipelines.

Before executing the NTRUEncrypt kernel, Altera OpenCL offline compiler compiled the kernel code into *.aocx file. Altera SDK for OpenCL creates several I/O interfaces such as memory controller to read and write data to external DDR memory and internal memory. On top of that, it creates the PCIe communication link for data communication and kernel code invocation between host and FPGA. A high-level representation of the OpenCL system implemented on FPGA can be seen in Fig. 2.

3.3 *Memory Hierarchy*

The OpenCL kernels have access to four distinct memory regions distinguished by access type and scope [7]. Global memory allows read-and-write access from all work-items across all work-groups. Local memory also provides read-and-write access to work-items; however, it is only visible to other work-items within the

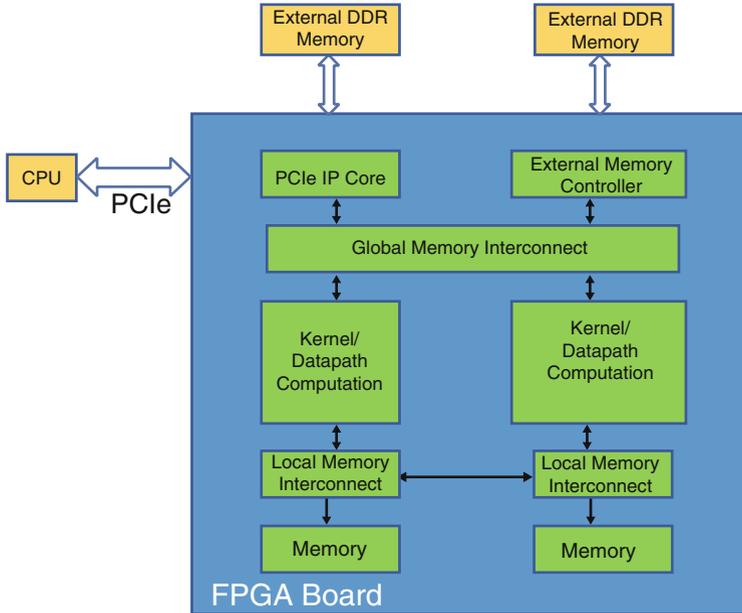


Fig. 2 OpenCL System on FPGA

same work-group. Constant memory is a region of read-only memory accessible to all work-items, thus immutable during kernel execution. Lastly, private memory provides read-and-write access visible to only individual work-items. Generally, its access speed is the fastest among the four memory regions.

NTRU encryption includes two kinds of data: plaintexts and key. In the first place, all the plaintexts and key are stored in the main memory of host. These data will be transferred into global or constant memory of the FPGA before execution. In order to maximize the performance, it is better to store these data in memory for the sake of speed. However, data features should be carefully considered during the memory distribution.

On FPGA, all local memory resources on FPGA are user-managed, and no explicit cache hierarchy exists, which is totally different from CPU and GPU. Offline compiler must generate complicated arbitration logic to deal with the memory access requests. One of the solutions to overcome memory bottleneck is to explicitly cache the elements of r and h in constant or local memory. Especially for r , this turns out to be a good strategy, since each $long$ contains 32 coefficients, thereby reducing the number of accesses to global memory with a factor 32.

4 Implementation and Optimization

4.1 Key Generation

Before invoking NTRUEncrypt key generation or encryption, a deterministic random byte generator (DRBG) should be instantiated. The DRBG in this project implements the ANS X9.82 Part 3-2007 standard, using HMAC_DRBG. The security level of DRBG should be equal to or greater than the security level of NTRU parameters. Thus, 256-bit levels have been adopted for DRBG. The DRBG instantiation function returns a handle that we can pass to the key-generation and encryption functions.

The selection of the NTRU PKC parameters defines the different levels of security. Also, p and q must have no common factors. To provide security level as high as possible, we choose the parameter set NTRU_EES743EP1, which is an IEEE 1361.1 parameter set that gives 256 bits of security and is a trade-off between key size and encryption/decryption speed. The equivalent security level of RSA is 15,360 bits and ECC is 512 bits.

Major parameters and operation of this step are listed as follows:

- The degree parameter: $N = 743$
- Modular: $p = 3, q = 2048$
- Create private key: the key pair (f, f_p)
- Create public key: $h \equiv f_q^{-1} * g \pmod{q}$.

The algorithm assumes $q = 2^w$ so the reduction will be performed by extracting the lower w bits. In the mean time, f is a randomly generated polynomial with small coefficients, and $f^{(-1)}$ is the multiplicative inverse of f and computed using the extended Euclidean algorithm in this work. The key setup and the random keys generation are done offline by the host.

Algorithm 1 NTRU key generation

```

1: KeyGen ( $N; g; q; p; h; f; f_p; f_q$ )
   Require:  $p, q, N$  and random polynomials,  $f$  and  $g$ .
2:  $Inverse\_Poly\_f_q(N; q; f; f_q)$ 
3:  $Inverse\_Poly\_f_p(N; q; f; f_q)$ 
4:  $PolyMultiply(f_q; g; h; N; q)$ 
5: for  $i = 0$  to  $N - 1$  do
6:   if  $h[i] < 0$  then  $h[i] = h[i] + q$ 
7:     Make sure all coefficients of  $h$  are positive.
8:   end if
9:    $h[i] = h[i] * p \pmod{q}$ 
10: end for
11: KeyGen returns the Public Key  $h$  and the polynomial inverse  $f_p$  through the argument list.

```

4.2 Encryption

As we mentioned in Sect. 2, the basic encryption operations of $e = r * h + m \pmod q$ are polynomial multiplication and addition. Benefited by the special form of the prime number p , modular reduction in this step can be realized using only shifts, additions, and subtractions. To encrypt n messages, each of size N elements, the polynomial multiplication between $r(x)$ and $h(x)$ is done first, and then the plaintext messages $m_i(x)$, $i = 1 \dots n$ are added mod q to the multiplication output.

The Algorithm 2 performs the polynomial multiplication of $r * h \pmod q$. As a reminder, the q in Step 4 is either p or q of Algorithm 1. It depends on which one is passed into the function. This algorithm only executes Step 4 if the current coefficients of $r[i]$ and $h[j]$ are both non-zero, which is in contrast with the guideline. This step approximately eliminates a third of unnecessary operations.

Algorithm 2 Polynomial multiplication in NTRUEncrypt

```

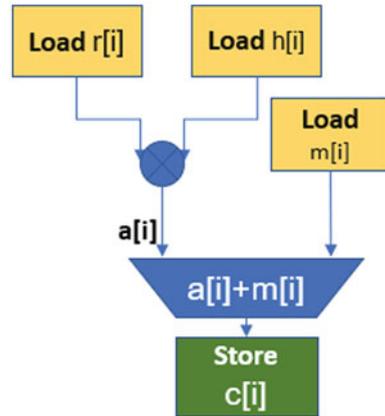
PolyMultiply ( $N; q; r; h$ )
Require:  $N$ , the coefficient modulus,  $q$ , and the two polynomials  $r$  and  $h$ .
Input:  $h = h_0, \dots, h_{N-1}$ .  $r = r_0, \dots, r_{N-1}$ 
Output:  $e = e_0, \dots, e_{N-1}$ 
1:  $e^{(0)} = 0$ 
2: for  $i=1$  to  $N$  do
3:   for all  $i = 0$  to  $N - 1$  do
4:      $e_i^{(j)} = e_{(i+1) \bmod N}^{(j-1)} + h_{(i+1) \bmod N} \times r_{(j-1)} \bmod q$ 
5:   end for
6: end for
7:  $e = e^{(N)} \bmod q$ 

```

4.3 Kernel Optimization

OpenCL is a strict framework that divides data into arrays and algorithms into kernel code that can manipulate this data. Therefore, the data permutation is a common operation used in various algorithms when implementing on FPGAs. Figure 3 illustrates the basic circuit-level structure for NTRU encryption. As a result, all operations of the encryption kernel are allocated dedicated hardware resources on the FPGA, prior to kernel execution. During execution, work-items step through each stage of the kernel one at a time. However, since each stage has dedicated hardware, multiple work-items may be passing through the circuit at any given moment, thus yielding pipeline parallelism. A key problem in designing this NTRU architecture is to permute streaming data. Permuting a long data sequence through hardware wiring leads to high area consumption and routing complexity. The kernel optimization techniques adopted in our implementation were introduced as follows:

Fig. 3 Simplified kernel operations of NTRU encryption system on FPGA



- (1) **Multiple kernel copies.** While OpenCL kernels are compiled to hardware logic circuits of fixed size, it is very common that a large portion of remaining FPGA resources are idling. To achieve the desired throughput, multiple copies of the NTRUEncrypt kernel were required. We create multiple copies of the NTRUEncrypt kernel pipelines to utilize the remaining resources of FPGA. These pipelines can execute independently from one another, and performance can scale linearly with the number of copies. Replication is handled in Altera OpenCL by setting the *num_compute_units* kernel attribute [8]. Fortunately, FPGAs are particularly efficient at integer arithmetic by allowing more than one work-group to fit within the FPGA. Targeting multiple work-groups is done using a simple kernel attribute *_attribute((num_copies(n))*.
- (2) **Loop unrolling and pipelining loop.** The removal of loop counter or loop testing logic is a benefit for the NTRUEncrypt algorithm on FPGAs. The nature of the NTRU encryption algorithm allows the entire code to be unrolled into a single very deep pipeline containing thousands of integer operations. The ACL compiler has *#pragma* directives that can be added to OpenCL code to instruct nested loops to be unrolled, allowing the full NTRUEncrypt code to be flattened. An optimally unrolled loop is a loop iteration that is launched every clock cycle. Launching one loop iteration per clock cycle maximizes pipeline efficiency and yields the best performance. As shown in the figure below, launching one loop per clock cycle allows a kernel to finish faster (Fig. 4).
- (3) **Using I/O channels.** Another optimization on our FPGA version not currently present on GPUs is to take advantage of I/O channels and kernel channels (OpenCL 2.0 pipes) [9]. As Fig. 5 shows, kernel channels allow encryption kernels to transfer data to one another via a first-in-first-out (FIFO) buffer, without the need for host interaction. Implementation of channels decouples data movement between concurrently executing encryption kernels from the host processor. Data written to a channel remains in a channel as long as the

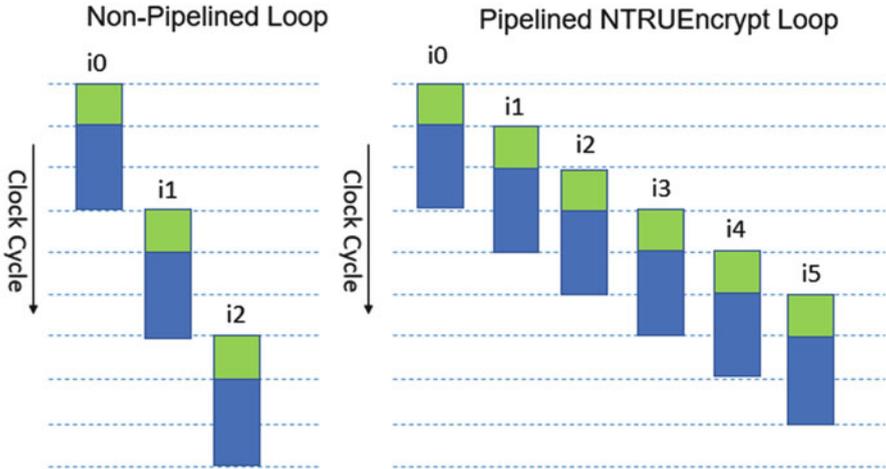


Fig. 4 The launch frequency of a loop iteration between a non-pipelined loop and a pipelined loop

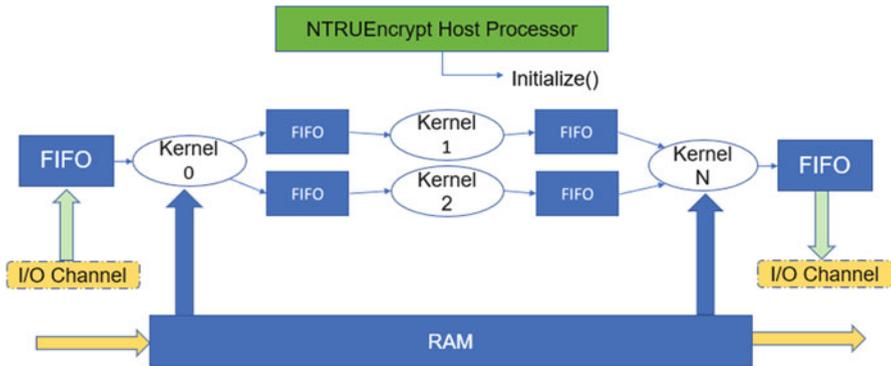


Fig. 5 An overview of channels' implementation

encryption kernel program remains loaded on the FPGA device. In other words, data written to a channel persists across multiple work-group invocations.

- (4) **Kernel vectorization.** Whereas replication makes extra copies of the encryption kernel pipeline, kernel vectorization maintains a single pipeline where each work-item then does N times as much work. By dealing with larger units of work, kernel vectorization can even reduce the number of loads and stores. In our encryption step, all kernels' arguments are uniform, and there are no return value while each computation is dedicated to its corresponding m .
- (5) **Changing the memory access pattern.** As we mentioned in Sect. 3, multiple memory interfaces have been configured on a single FPGA board. The NTRU-Encrypt kernel performs a large number of memory accesses. Therefore it is

important to direct which interface should be used for individual buffers, which can be done via attributes.

Our experimental broad uses SDRAM as global memory, which was configured as burst-interleaved in default. In most circumstances, the default burst-interleaved configuration leads to the best load balancing between the memory banks. However, in our case, we partition the banks manually as two non-interleaved and contiguous memory regions to achieve better load balancing [7]. Contiguous memory access optimizations analyze statically the access patterns of global load and store operations in a kernel. By basing the array index on the work-item global ID, the offline compiler can direct contiguous load operations. As shown in Fig. 6, load operations retrieve the data sequentially from the input array and send the read data to the pipeline as required. Contiguous store operations then store elements of the result that exits the computation pipeline in sequential locations within global memory.

Our another optimization of memory access efficiency is to minimize the number of global memory accesses by preloading the data from a group of computations from global memory to constant memory. Constant memory resides in global memory, but the kernel loads it into an on-chip cache shared by all work-groups at runtime. However, unlike global memory accesses that have extra hardware to tolerate long memory latencies, the constant cache suffers large performance penalties for cache misses. As the security parameter grows, the preloaded data r and h in the constant buffer cannot fit into the constant cache. We achieve better performance with *_global const* arguments instead of *_constant*. In this way, if the host application writes to constant memory that is already loaded into the constant cache, the cached data is discarded (i.e., invalidated) from the constant cache.

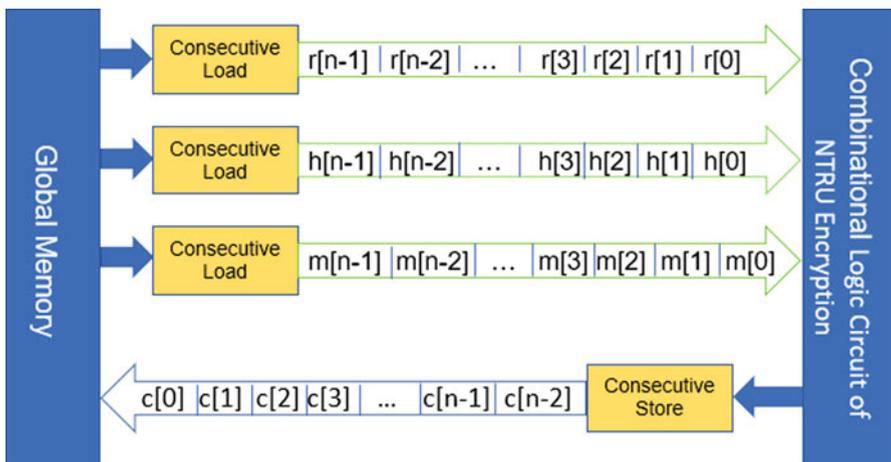


Fig. 6 Contiguous memory access

5 Experimental Results And Analysis

5.1 Experimental Setup

In this part, we detail comparison between FPGA and GPU implementation versions in terms of throughput and power consumption. We accelerate the running time of these cryptosystems by exploiting the inherent parallelism in computations through an FPGA-based parallelism implementation.

The NTRUEncrypt system is implemented on Intel i7-9700K CPU with OpenCL-enabled Altera Stratix V FPGA with 16 GB DDR3, 50 MHz oscillator, and 2 PCI Express hard IP blocks that support for PCIe Gen1/2/3. Intel OpenCL 2.0 development kit and Altera Quartus 18.0 are also used. The parallel implementation is based on the OpenCL framework and can run on arbitrary hardware platform accelerators with minor changes. To measure the performance improvement, the same OpenCL source code with minor change was compiled and ran on an NVidia 2080 GPU card with 1515 MHz clock rate.

5.2 Comparisons and Analysis

Our experiments illustrate the power consumption and FLOP throughput for the GPU-CPU and FPGA-CPU implementations (Table 1). The FPGA parallelism version performs as expected: In GPU, the throughput for the same security level is 72 MB/s. In contrast, FPGA modules speed up encryption by a factor of nearly 1.5. The average encryption throughput of FPGA is 113 MB/s. The power consumption of the FPGA accelerator is also significantly lower, requiring approximately 55 watts compared to several hundred watts on the GPU. To achieve 113 MB/s throughput for the NTRU encryption described here, only 54% of the Stratix V FPGA device was utilized. The remainder could be left unused for power savings, or extra kernels could be further optimized and placed in parallel to the encryption core.

By focusing hardware resources only on the algorithm to be executed, FPGAs can provide better performance per watt than GPUs for this work. The key difference between kernel execution on GPUs and FPGAs is how parallelism is handled. GPUs are “single-instruction, multiple-data” (SIMD) devices – groups of processing elements perform the same operation on their own individual work-items. On the

Table 1 Performance comparison of NTRUEncrypt in FPGA and GPU

	Encryption throughput	Power consumption
Stratix V FPGA	113 MB/s	55 watts
GTX 2080 GPU	72 MB/s	~230 watts

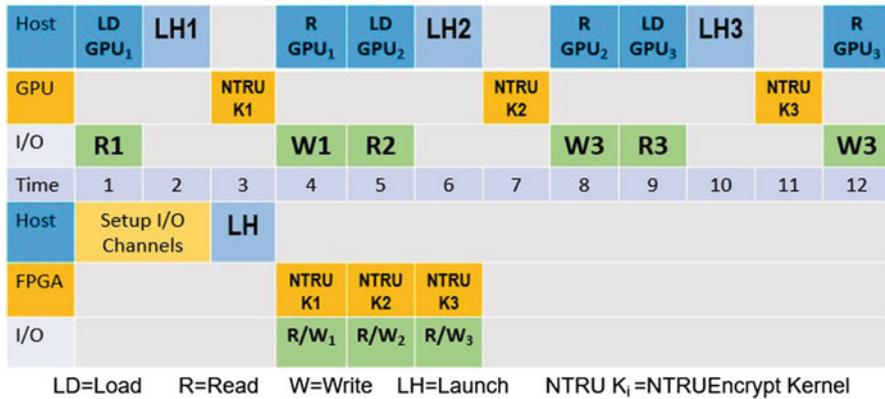


Fig. 7 FPGA vs GPU: I/O channel benefits

other hand, FPGAs exploit pipeline parallelism – different stages of the instructions are applied to different work-items concurrently.

GPUs consist of hundreds of simple processing “cores,” which can each handle their own work-items. Several GPU cores execute the same instruction in lock-step with one another as a SIMD unit of fixed size (sometimes called a warp). On an FPGA, each NTRUEncrypt kernel is compiled to a custom circuit.

As we mentioned in Sect. 4, the I/O channels and kernel channels make FPGA stronger than GPU. Traditionally, GPU kernels that want to pass data to one another may do so by issuing reads and writes to global memory combined with synchronization. Performance and power efficiency gains are achieved by the removal of these intermediate reads and writes. FPGAs extend the idea of kernel channels even further to allow I/O pipes, which allow kernels to access directly from a streaming interface without host interactions, known as I/O channels. Effectively the NTRUEncrypt host configures the data pipeline and then steps out of the data path. Figure 7 illustrates an NTRU encryption kernel being executed on three sets of data coming from an I/O source. Significant time savings are achieved because the FPGA communicates directly with the I/O source and no longer needs the host to serve as a middleman.

This FPGA implementation takes advantage of computing unit replication, kernel SIMD vectorization, and non-blocking I/O channels to achieve higher throughput and lower kernel time. The SIMD vectorization duplicates only the data path of the compute unit without generating additional memory interfaces. When the kernel is vectorized, the static memory coalescing is performed automatically by the compiler to generate a memory interface that can coalesce the multiple memory loads into a single wide load.

6 Summary

The general purpose of the implementation presented herein was to provide a reference FPGA implementation of the NTRU encryption scheme and to assess its real-world properties. The operations of the NTRU encryption algorithm show good characteristics of parallel processing which makes NTRU a good candidate to benefit from the high degree of parallelism available in FPGA. FPGAs offer a middle ware among the platforms with high programmability and energy efficiency without sacrificing the throughput of the NTRUEncrypt. It is also observed that the FPGA performs better than GPU as a pipeline complexity grows. NTRUEncrypt gives incredible performance gains at no loss in security on an FPGA-based heterogeneous system. There are areas where significant improvements can be made and fine-tuned. Furthermore, the OpenCL specification can make FPGAs even more useful.

References

1. D. Micciancio, O. Regev, Lattice-based cryptography, in *Post-quantum Cryptography* (Springer, Berlin/Heidelberg, 2009), pp. 147–191
2. J. Hermans, F. Vercauteren, B. Preneel, Speed records for NTRU, in *Cryptographers' Track at the RSA Conference* (Springer, Berlin/Heidelberg, 2010)
3. T. Bai et al., Analysis and acceleration of NTRU lattice-based cryptographic system, in *15th IEEE/ACIS SNPD* (IEEE, 2014)
4. J. Hoffstein, J. Pipher, J.H. Silverman, NTRU: a ring-based public key cryptosystem, in *International Algorithmic Number Theory Symposium* (Springer, Berlin/Heidelberg, 1998)
5. J. Hoffstein, J. Silverman, Optimizations for NTRU, in *Proceedings of the Conference of Public-Key Cryptography and Computational Number Theory* (2001)
6. J. Hoffstein et al., Practical lattice-based cryptography: NTRUEncrypt and NTRUSign, in *The LLL Algorithm* (Springer, Berlin/Heidelberg, 2009), pp. 349–390
7. Intel FPGA SDK for OpenCL Pro Edition: Programming Guide. 2019.12
8. H.R. Zohouri, High performance computing with FPGAs and OpenCL. arXiv preprint arXiv:1810.09773 2018
9. Intel FPGA SDK for OpenCL Pro Edition: Best Practices Guide. 2019.9

High-Performance and Energy-Efficient FPGA-GPU-CPU Heterogeneous System Implementation



Chaoyu Zhang, Hexuan Yu, Yuchen Zhou, and Hai Jiang

1 Introduction

In 2019, Intel introduced the oneAPI programming model which provides a comprehensive and unified portfolio of developer tools that can be used across hardware and applications to take advantage of the oneAPI programming model. This latest programming model can execute on multiple target hardware platforms ranging from CPU, GPU, and FPGA [1]. Thus, selecting the most suitable hardware architecture based on the diversity and pattern of different applications to enhance power efficiency and performance needs to be deeply discussed.

FPGA accomplishes applications through hardware logic design, which is reconfigurable integrated circuit, for unique characters and advantages. FPGA is a balancing act between ASICs and general-purpose processors [2]. On the one hand, comparing with CPU or GPU, the hardware programming makes it more power efficient than general-purpose processors at the cost of lower flexibility and high complexity of software programming [3]. As the execution of applications is based on pre-designed hardware logic rather than separate instructions, FPGA can directly be connected to inputs and can offer very high bandwidth with lower latency. On the other hand, the reconfigurability provides more flexibility than ASICs and lower developmental cost as well as shorter periods [4].

FPGA architecture is composed of CLBs (configure logic blocks), programmable routing, and programmable I/O cells as shown in Fig. 1; CLBs are composed of SRAM (static random access memory) cells in the form of loop-up tables (LUTs) to implement combinational and sequential logic; programmable routing architecture

C. Zhang (✉) · H. Yu · Y. Zhou · H. Jiang

Department of Computer Science, Arkansas State University, Jonesboro, AR, USA

e-mail: chaoyu.zhang@smail.astate.edu; hexuan.yu@smail.astate.edu;

yuchen.zhou@smail.astate.edu; hjiang@astate.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_35

477

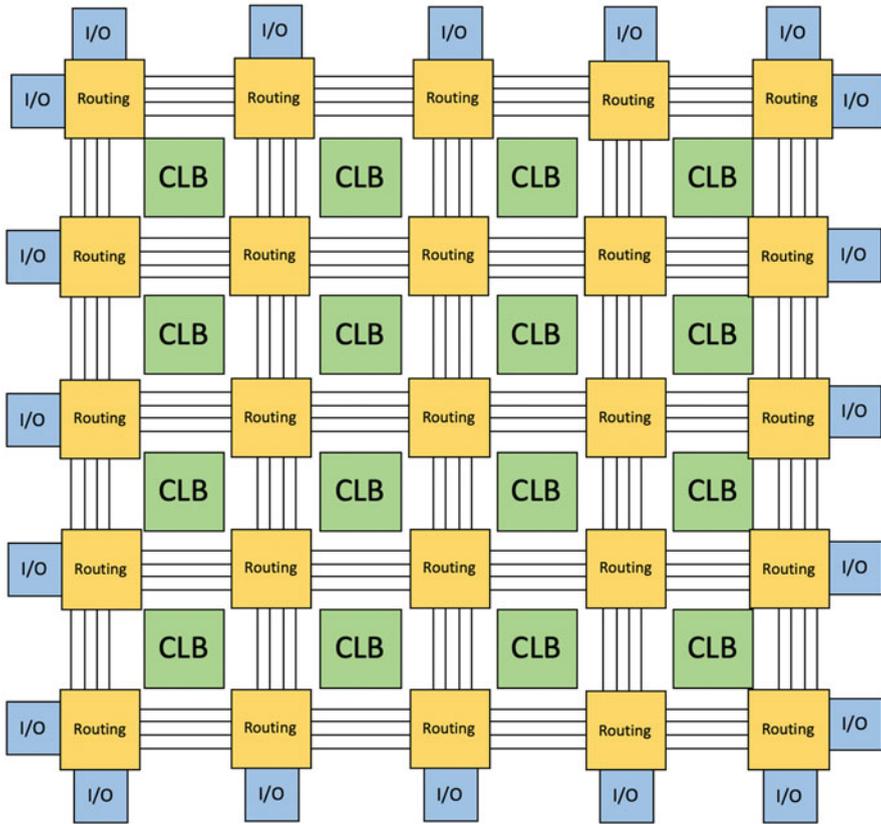


Fig. 1 FPGA high-level architecture

provides a self-defined interconnection between CLBs and also gives routing connections among logic blocks and I/O blocks to complete a fully functional circuit; and I/O cells are important components connecting the external peripherals and controlling the communications.

Thus, FPGA can be reconfigured to implement logic by reassigning the content of the LUTs and reconnecting the routing configuration. Modern FPGAs add more mature and efficient modules like DSP (digital signal processors), large memory blocks (block RAMs), and different I/O controllers (DDR, PCIe, network, etc.). These components enhance general purpose computing and save logic blocks implementing LUTs [5, 6].

GPU can offer good peak performance and high memory bandwidth. They have been widely deployed in high-performance computing (HPC) systems as accelerators. The multiprocessors employ a SIMT (single instruction multiple threads) to manage hundreds of threads. Each thread is mapped into one SP core and executes independently with its own instruction address and register state.

The Nvidia GPU has a multilevel memory hierarchy. The types of memory can be classified as global memory, shared memory, and registers. The effective bandwidth of each type of memory depends significantly on the access pattern. Global memory is relatively large but has a much higher latency compared with the on-chip shared memory. The shared memory is on-chip memory, much faster than the global memory, but we also need to avoid the problems of bank conflict and limited size. After Nvidia Kepler architecture introduces a new warp-level intrinsic called the shuffle operation, it allows the threads of a warp to exchange data with each other directly by operating registers of threads without going through shared (or global) memory [7]. The shuffle instruction has a lower latency than shared memory access and does not consume shared memory space for data exchange, so this can present an attractive way for applications to rapidly interchange data. Threads are organized in warps. A warp is defined as a group of 32 threads of consecutive thread IDs. These warps of threads are organized into thread blocks. Thread blocks are distributed among SMs and split into warps scheduled by SIMT units. All threads in the same thread block share the same shared memory and can synchronize themselves by a barrier. Threads in a warp execute one common instruction at a same time [8, 9].

In this paper, our goal is select the most suitable hardware architecture based on the diversity and patterns of different applications to enhance power efficiency and performance. With the support of OpenCL for FPGA and CUDA for GPU, we schedule different kinds of workloads on specific accelerators. Figure 2 shows the overview of an heterogeneous system consisting of different processing elements including the CPU, GPU, and FPGA. The main feature of this architecture is using current computing platforms with FPGA and GPU that communicate via PCIe within the system. Therefore, each application can be deployed onto a certain accelerator for the sake of high performance and minimized power consumption. However, choosing a proper processing element for a specific workload, with maximum performance and minimum energy consumption, needs more experiments.

To illuminate this area, we examined the performance and energy consumption of processing units such as CPU, GPU, and FPGA, with the Rodinia Benchmark Suite, version 3.1 [10] and CUDA code samples [11]. Rodinia is designed for heterogeneous computing infrastructures. A vision of heterogeneous computer systems that incorporate diverse accelerators is widely shared among researchers and many industry analysts. Programming model overview is shown in Fig. 3 (OpenCL host and OpenCL kernel). The host code is for programming a host application running on a host PC to manage an FPGA device at runtime with a set of common API (application programming interface) and is compiled using a standard C compiler to generate a host binary. The kernel code is launched to FPGA or GPU. As for the FPGA kernel, it is compiled with the Intel FPGA OpenCL compiler and converted into synthesizable Verilog HDL files. Then aocx files with FPGA configuration information are generated by Intel Quartus Prime. The aocx file is downloaded to FPGA at runtime of the application on host through APIs, and the input data required for the kernel and the output resulting data are transferred via PCIe bus [4, 12].

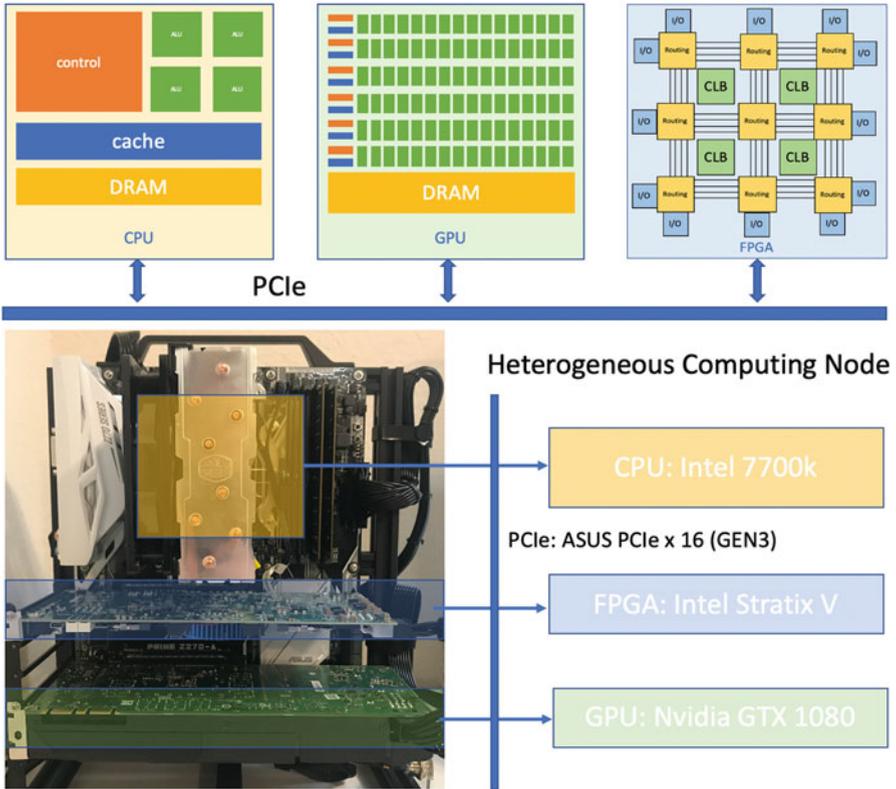


Fig. 2 Heterogeneous programming composed of CUDA and OpenCL

Most recent research focuses on a single accelerator, such as FPGA and GPU. The integration of GPU and FPGA hasn't been addressed often. Although several studies have attempted to explore this issue, related FPGA implementations are based on hardware description languages such as VHDL and Verilog at quite low-level. Therefore, task-level scheduling task characteristics over FPGA-GPU-CPU heterogeneous architecture are the biggest difference between our work and previous research.

The remainder of this paper is organized as follows: Sect. 2 gives an introduction to FPGA-GPU architectures. Section 3 describes the FPGA-GPU programming model. Section 4 introduces the FPGA-GPU benchmark kernels. Section 5 describes the FPGA-GPU heterogeneous implementation and result analysis. Finally, concluding remarks are stated in Sect. 6.

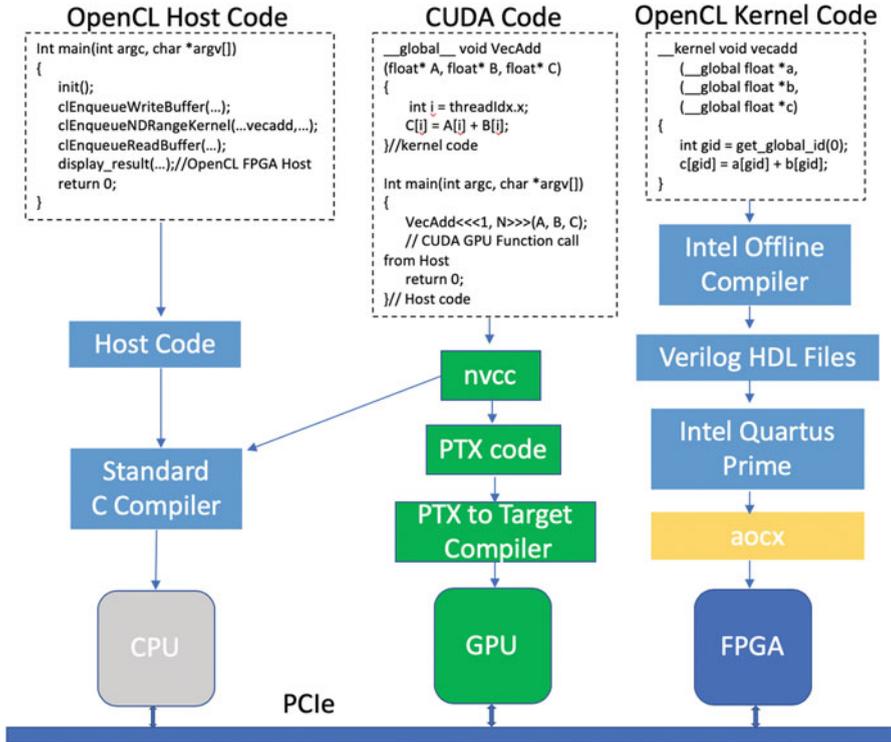


Fig. 3 FPGA OpenCL and GPU CUDA programming model

2 FPGA-GPU-CPU System Architectures

2.1 FPGA

Comparing FPGA with general-purpose processors such as CPU or GPU, the hardware programming makes it more power efficient than general-purpose processors with a cost of lower flexibility and high complexity of software programming. Since the application execution is based on pre-designed hardware logic rather than separate instructions, FPGA can directly be connected to inputs and offer very high bandwidth with lower latency. Moreover, the reconfigurable characters provide more flexibility than ASICs with a cost of lower development investment and shorter development period. However, FPGA programming is not as simple as C programming used in general-purpose processors. To create an FPGA design, the application is processed by Intel OpenCL SDK. Eventually, an FPGA bitstream is created. One of the most time-consuming processing is synthesizing hardware description into a netlist, which is just a “list of nets,” connecting gates or flip-flops

together. With Intel Stratix V FPGA, the total synthesis time is up to several hours, including the time to establish a large scale routing connections [13].

Intel Stratix V FPGA is capable to be deployed as an accelerator, as shown in Fig. 4. In this FPGA, the “soft-logic” consists of ALMs (Adaptive Logic Modules) with multiple CLBs, and the “hard-logic” consists of DSPs, block RAMs, multiple controllers, transceivers, and phase-locked loops (PLL). In the Stratix V FPGA, each ALM consists of multiple-input LUTs, adders and carry logic, and registers (Flip-Flops). Each adaptive LUT is capable of implementing multiple combinations of different functions including one 6-input function.

The DSP block in Intel Stratix V FPGA is designed to implement 9-bit, 18-bit, 27-bit, and 36-bit word lengths fully registered addition, multiplication, and fused multiply and add operation; one 27-bit-by-27-bit integer or fixed-point multiplication; data cascading, fast Fourier transform; and multiple several filters.

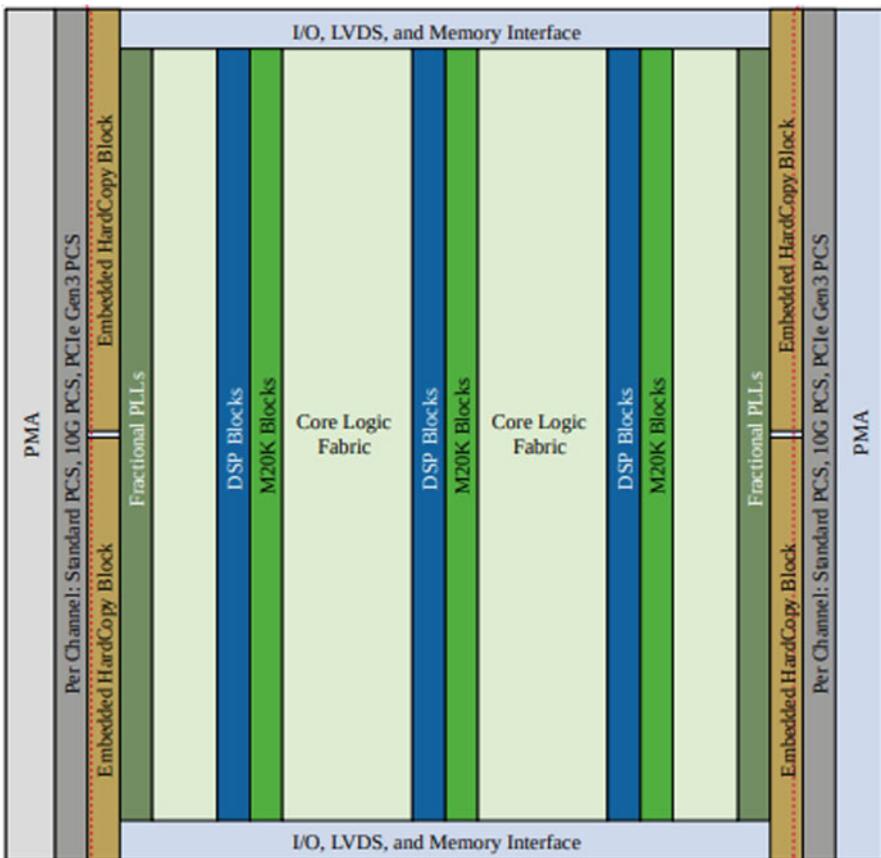


Fig. 4 Intel Stratix V FPGA architecture

Furthermore, multiple DSPs can be chained to implement dot products or other complex operations.

M20K block is RAM module in the Intel Stratix V device, capable of storing a maximum of 20K bits of data. Each block has two independent ports for bidirectional read and write. Data can be stored in each block up to 40-bit word-length with 9-bit memory address. Furthermore, M20K is suitable to implement first-in first-out buffers (FIFO) or shift registers. Arrays of M20K blocks could be built as buffers [14].

2.2 GPU

The GPU architecture is built with an array of multi-threaded SMs (streaming multiprocessors), within each of SM composed of SP (scalar processor) cores. The multiprocessors employ a single instruction multiple threads (SIMT) model to manage hundreds of threads. Each thread is mapped into one SP core and executes independently with its own instruction address and register state.

GPU devotes more transistors to data processing. This conceptually works for highly parallel computations because GPU can hide memory access latencies with computation instead of avoiding memory access latencies through large data caches and flow control. Data-parallel processing maps data elements to parallel processing threads. Many applications that process large data sets can use a data-parallel programming model to speed up the computations.

The Nvidia GPU has a multilevel memory hierarchy. The types of memory can be classified as global memory, shared memory, and registers. The effective bandwidth of each type of memory depends significantly on the access pattern. Global memory is relatively large but has a much higher latency compared with the on-chip shared memory. Global memory is not cached, so it is important to follow the right access pattern to achieve good memory bandwidth. Threads are organized in warps. A warp is defined as a group of 32 threads of consecutive thread IDs. A half-warp is either the first- or second-half of a warp. The most efficient way to use the global memory bandwidth is to coalesce the simultaneous memory accesses by threads in a half-warp into a single memory transaction [8]. Since it is on chip, the shared memory is much faster than the global memory, but we also need to avoid the problems of bank conflict. After Nvidia Kepler architecture introduce a new warp-level intrinsic called the shuffle operation, it allows the threads of a warp to exchange data with each other directly by operating registers of threads without going through shared (or global) memory. The shuffle instruction has a lower latency than shared memory access and does not consume shared memory space for data exchange, so this can present an attractive way for applications to rapidly interchange data.

Programming Nvidia GPU for general-purpose computing is supported by the Nvidia CUDA (Compute Unified Device Architecture) environment. CUDA programs on the host (CPU) invoke a kernel grid, which runs on the device (GPU). The same parallel kernel is executed by many threads. These threads are organized

into thread blocks. Thread blocks are distributed to SMs and split into warps scheduled by SIMT units. All threads in the same thread block share the same shared memory of size 48 KB and can synchronize themselves by a barrier. Threads in a warp execute one common instruction at a time. This is referred to as warp-level synchronization. Full efficiency is achieved when all 32 threads of a warp follow the same execution path. Branch divergence causes serial execution.

3 FPGA-GPU Programming Model

In this heterogeneous computing node, we use a mixed compilation solution to cooperate different processing elements. Figure 5 shows that the flow of the compilation was implemented in multilingual programming composed of CUDA and OpenCL, and therefore separate compilation was needed. The CUDA code and OpenCL host code were compiled with `nvcc` and `g++` separately, and the generated object files were linked using `nvcc` to generate an executable and linkable format (ELF) file. The OpenCL kernel code for the computation was compiled offline using the Intel FPGA OpenCL compiler [15].

3.1 Intel FPGA SDK for OpenCL

OpenCL is an open-source standard for programming heterogeneous systems. The OpenCL-based applications consisted of the host code that executes on the host CPU and can be written in compatible high-level language and C-based device kernel code. It provides APIs for controlling the accelerator and communicating

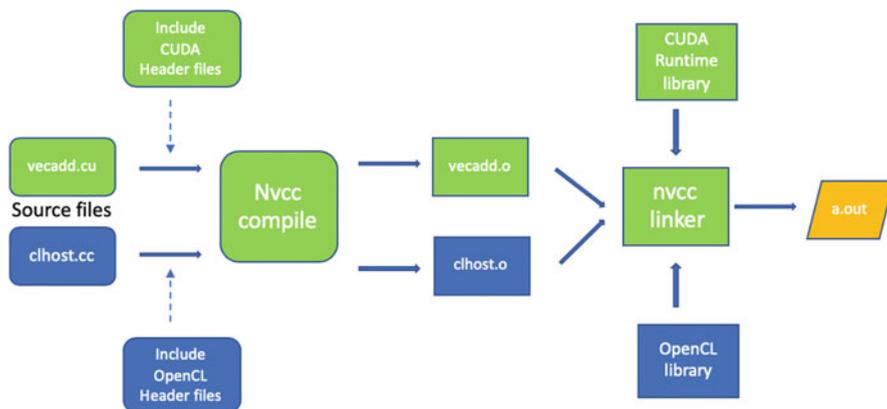


Fig. 5 Flow of heterogeneous compilation

between the host processor and the accelerator. The typical work flow of OpenCL computation is allocating host memory for input data and transferring data to the FPGA device memory by PCIe bus; once transferring is completed, FPGA kernel is launched for computation. After that, output data are sent back to the host memory [16].

In OpenCL, each thread is called a work-item and multiple work-items are grouped to form a work-group. To execute an application, the thread space is distributed over multiple work-groups. Within each work-group, work-items are synchronized using barriers, and data can be shared between the work-items using the fast on-chip local memory. However, the only way to share data between different work-groups is through the slow off-chip memory. The number of work-items in a work-group is called the local work size, and the total number of work-items necessary to fully execute an application is called the global work size. Work-items and work-groups can be arranged in multiple dimensions, up to three, in an index space called an NDRange [17].

In OpenCL, multiple memory types are defined. Global memory is inserted on the FPGA board with largest memory size but much slower. It can be accessed by all work-items of all work-groups. Global memory consistency is only guaranteed after a kernel is executed completely. Local memory is on-chip memory of the device and can be used to share data between the work-items within a work-group. Each work-group has its own local memory space, the local memory space of a work-group cannot be accessed by other work-groups, and local memory consistency is guaranteed by barriers. Constant memory is an on-board and read-only memory for fast data access. Private memory belongs to each work-item as registers with very limited size [18].

Intel FPGA SDK for OpenCL provides the necessary APIs and runtime to program and use PCIe or system-on-chip (SoC) FPGA similar to GPU or other accelerators. The necessary IP cores communicate between the FPGA, external DDR memory, and PCIe, alongside with necessary PCIe and DMA drivers for communication between the host and the FPGA provided by the board manufacturers in form of a board support package (BSP). It relieves the programmer from the burden of having to manually set up the IP cores and create the drives. It is designed with traditional HDL-based FPGA designs. Some BSPs also provide the possibility to send and receive data using FPGA on-board network ports. Runtime compilation of OpenCL kernels is not possible for FPGA due to very long placement and routing time. Therefore, the OpenCL kernel needs to be compiled offline into an FPGA bitstream and then loaded at runtime by the host code to reprogram the FPGA and execute the application [15, 16].

3.2 *Nvidia nvcc for CUDA*

The source files for CUDA applications consist of a mixture of conventional codes. CUDA compilation separates the device functions from the host code, compiles

the device functions using proprietary Nvidia compilers or assemblers, compiles the host code using a stander C compiler that is available on the host platform, and afterward embeds the compiled GPU functions as load images in the host object file. In the linking stage, specific CUDA runtime libraries are added to support remote SIMD procedure calls and provide explicit GPU manipulation and generate object files including OpenCL library for FPGA linked by nvcc to generate an executable and linkable format file.

This compilation involves several splitting, compilation, preprocessing, and merging steps for each CUDA source file. Finally, nvcc linker is used to connect object files of FPGA-GPU together. It is the purpose of the CUDA compiler driver nvcc to hide the intricate details of CUDA compilation from developers and be more convenient for programmers [8, 9].

4 FPGA-GPU Heterogeneous Kernels

Multiple benchmark suites have been proposed as representatives of HPC applications to evaluate different hardware and compilers. We can take advantage of the existing OpenMP and CUDA implementations for evaluating CPU and GPU. Furthermore, we port and optimize FPGA kernels based on Intel OpenCL SDK to figure out the meaningful performance and energy efficiency comparison between different hardware architectures and show the strengths and weaknesses of different hardware devices. We implement and analyze several famous widely used kernels of Rodinia Benchmark Suite 3.1 and port CUDA code samples to FPGA, expecting our FPGA-GPU specific scheduling for workloads has unique advantages over CPUs or CPU-GPU system. In this study, the heterogeneous system consisted with the Intel i7-7700k CPU, the Nvidia GTX-1080 GPU, and the Intel Terasic Stratix-V GX FPGA.

The benchmarks tested on Intel FPGA and Nvidia GPU for performance and energy efficiency analysis are listed as follows:

- **MM**: Matrix multiplication is based on CUDA sample matrix multiplication, computing $C = A * B$ in parallel [11].
- **FFT**: Fast Fourier transforms (FFTs) are exploited in a wide variety of fields ranging from computer science to natural sciences and engineering. With the rising data production bandwidths of modern FFT applications, judging best which algorithmic tool to apply can be vital to any scientific endeavor [11].
- **NW**: Needleman-Wunsch is a nonlinear global optimization method for DNA sequence alignments. The potential pairs of sequences are organized in a 2D matrix [10].
- **GE**: Gaussian elimination computes result row by row, solving for all of the variables in a linear system. The algorithm must synchronize between iterations, but the values calculated in each iteration can be computed in parallel [10].

Name	Dwarfs	Area
Matrices Multiplication	Dense Linear Algebra	Linear Algebra
Fast Fourier Transforms	Spectral Methods	Digital Signal Processing
Needleman-Wunsch	Dynamic Programming	Bioinformatics
Gaussian Elimination	Dense Linear Algebra	Linear Algebra
Back Propagation	Unstructured Grid	Pattern Recognition
Speckle Reducing Anisotropic Diffusion	Structured Grid	Image Processing

Fig. 6 Summary of tested kernels

- **BP:** Backpropagation is a machine learning algorithm that trains the weights of connecting nodes on a layered neural network. The application’s activations are propagated from the input to the output by layer, and the error between the observed and requested values in the output layer is propagated backward to adjust the weights and bias values. In each layer, the processing of all the nodes can be done in parallel [10].
- **SRAD:** Speckle reducing anisotropic diffusion is a diffusion method for ultrasonic and radar imaging applications based on partial differential equations (PDEs). It is used to remove locally correlated noise, known as speckles, without destroying the important image features [10].

Figure 6 here is the summary of implementations.

5 Implementation and Results Analysis

We shall start our experiment with a group of performance comparison between CPU and other accelerators. Our results are the accelerator speedups (CPU running time is divided by accelerator running time) to demonstrate that no single architecture is the best for all workloads due to the incredible diversity. In Fig. 7, most of the cases except FFT show GPU has the best performance, comparing with 9.0 times speedup over CPU in general. In NW, BP, and SPRAD, the computing speedups of FPGA and GPU are basically at the same level but slightly inferior. However, in MM and GE, GPU computing speed is much faster than the other cases. One exception is FFT operation speedup on FPGA which is much higher than the rest of all hardware devices.

Due to the unique customized deep pipeline of FPGA, which successfully avoid instruction overhead in most of the other general-purpose accelerators with inserted pipeline pragmas into target loops, programmers can implement pipeline in FPGA more easily. To eliminate data dependency, we need to process the data in advance

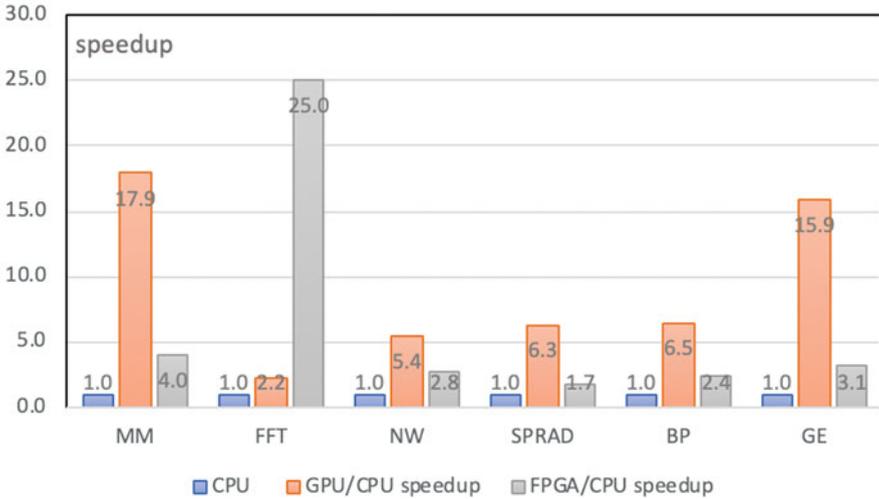


Fig. 7 Single accelerator performance speedup comparison

so that it is split into data blocks that can be calculated in parallel. Pipeline interval and depth are two key factors for measuring pipeline performance. They represent the number of cycles between the start of two consecutive iterations and the processing of the entire data iteration. The smaller the interval, the higher the pipeline throughput, the greater the depth, the higher the achievable speed [17, 18].

Another reason is that multiple programmable logic modules can perform calculations independently and simultaneously. This is similar to current multi-core and SIMD technologies. But related to SIMD technology, FPGA concurrency can be performed between different logic functions, not limited to performing the same function at the same time. That's why FFT has the best performance and dynamic programming. Unstructured grid kernels also yield speedup almost as same level as GPU.

In Fig. 8, it is obvious that FPGA has significant power efficiency advantages over any other processors. Therefore, in every benchmark, Stratix V FPGA can achieve higher power efficiency than GTX 1080 GPU as well as Intel i7 7700k CPU. The largest power efficiency advantage was observed in the GE benchmark test, that is, the power efficiency of Stratix V FPGA was 5.6 times of the same generation CPU. In MM, FPGA shows the great power efficiency amount GPU and FPGA, which is 6.3 times better than GPU. Because the compiled hardware circuit is used instead of executing instruction by instruction, FPGA has relatively stable and more efficient energy performance.

The general ratio of FPGA and GPU in computing speed as well as power efficiency is shown in Fig. 9. FPGA has more advantages in energy consumption in different examples, and GPU has better performance in a large number of parallel data calculations.

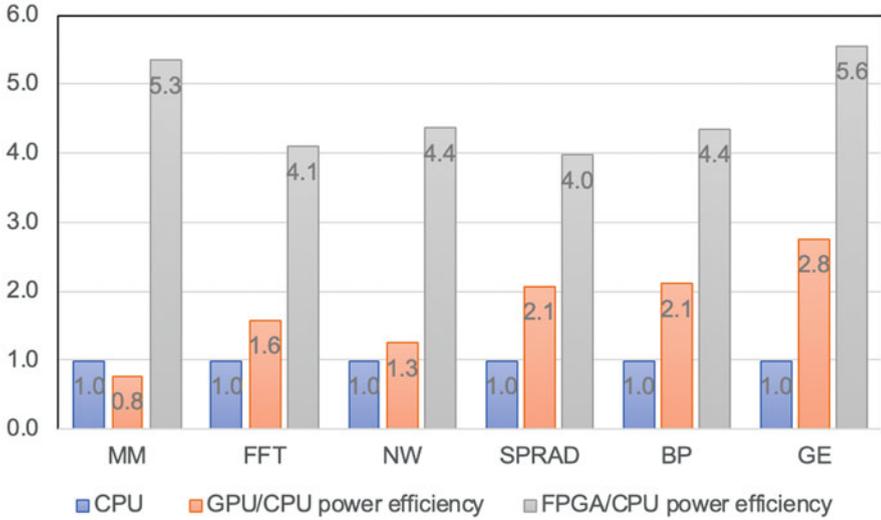


Fig. 8 Single accelerator power-efficiency comparison

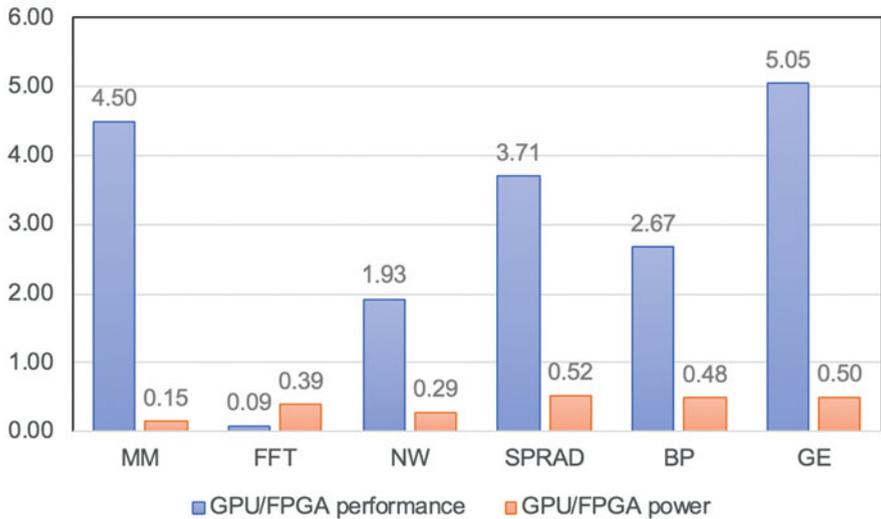


Fig. 9 Speedup and power efficiency of FPGA and GPU comparison

In order to enable different applications to achieve better computing performance or better energy efficiency on different hardware accelerators, we propose a high-performance-oriented scheduling scheme and an energy-efficient scheduling scheme. A high-performance scheduling scheme means that if this task cannot perform faster on the FPGA than the performance of the GPU or CPU, then

all subsequent tasks of this type are performed on the GPU. An energy-efficient scheduling scheme means that if the computational energy efficiency of this task on the GPU cannot be better than that of the FPGA, then such tasks will be executed on the FPGA. In Fig. 10, the optimal computing performance cannot be achieved in either the single GPU or CPU or FPGA. The performance-oriented scheduler launches the FFT kernel to FPGA and the rest of the benchmarks to GPU for better performance. This could be 12.8 times faster than all running on the CPU. In our proposed energy-efficient scheduling, all benchmarks run on FPGA. Since no accelerator can achieve such energy efficiency result as FPGA does. This also proves that FPGA has huge advantages in energy efficiency, and it is 4.6 times faster than single CPU scheduling. Especially when low-latency, low power consumption, and real-time large-data operations are required, this advantage could be unique and irreplaceable. In the past few years, the slowdown of Moore's law has made heterogeneous systems to be the breakthrough that has the potential to achieve better performance and energy efficiency [19, 20]. A common heterogeneous system consists of CPU and GPU as the most widely used combo. GPU has high performance across multiple application domains with rich software ecosystem enabling programmers to adopt them without facing many programmable barriers. Compared with other accelerators, FPGA can provide better power efficiency. On the downside, developing applications on FPGA requires hardware design knowledge, which is often the main obstacle to programmers. To alleviate this problem, advanced comprehensive frameworks using languages

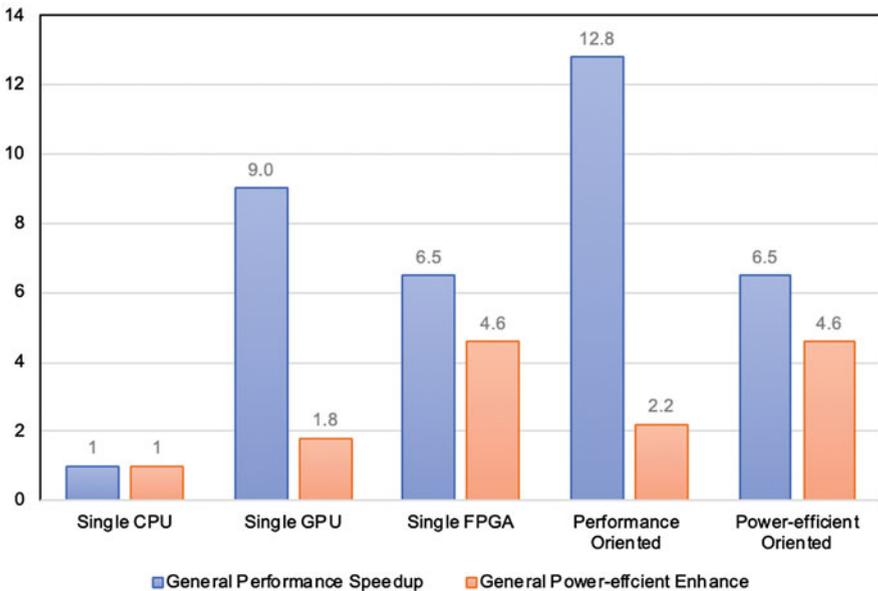


Fig. 10 General speedup and power efficiency of different scheduling

such as OpenCL have emerged to increase programmer usability. Therefore, it is far-reaching practical significance to design and explore deeper FPGA-GPU-CPU heterogeneous systems and to schedule among different hardware according to the characteristics of tasks.

6 Conclusion

In this paper, we first designed an alternative system composed of GPU, FPGA, and CPU. On top of this, we analyzed the hardware composition of the system and the different characteristics of hardware in terms of efficiency and high performance. After that, we introduced a feasible programming model that can run and schedule our assigned tasks on different hardware, choosing the most suitable hardware architecture for each application. On the FPGA-GPU-CPU system, high-performance-oriented and high-energy-efficiency-oriented working kernel scheduling are realized. This heterogeneous system can achieve better efficiency and higher performance than a single computing unit.

Intel proposed OneAPI in 2019. In order to make it easier to use heterogeneous accelerated systems across hardware, this includes CPU, GPU, and FPGA. In the future, we are likely to see supercomputers composed of these three. We hope to take this opportunity to explore the high performance and high efficiency of heterogeneous computing as an early exploration of resource management and dynamic scheduling and even some hardware virtualization in a system composed of FPGA-GPU-CPU. In summary, all in all, more diverse heterogeneous systems are worthy of our further research in order to break the slowdown of Moore's law and optimize energy efficiency.

References

1. Intel® oneAPI Programming Guide (Beta), <https://software.intel.com/en-us/oneapi-programming-guide>
2. H. R. Zohouri, N. Maruyama, A. Smith, M. Matsuda, S. Matsuoka, Evaluating and optimizing opencl kernels for high performance computing with FPGA, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, ser. SC'16* (IEEE Press, Piscataway, 2016), pp. 35:1–35:12. [Online]. Available: <http://dl.acm.org/citation.cfm?id=3014904.3014951>
3. M.P. Véstias, H.C. Neto, Trends of CPU, GPU and FPGA for high-performance computing, in *2014 24th International Conference on Field Programmable Logic and Applications (FPL)* (2014), pp. 1–6
4. R. Kobayashi, N. Fujita, Y. Yamaguchi, A. Nakamichi, T. Boku, GPU-FPGA heterogeneous computing with OpenCL-enabled direct memory access, in *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)* (2019)
5. I. Kuon, R. Tessier, J. Rose, FPGA Architecture (2008)

6. Alter FPGA Architecture White Paper, <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/wp/wp-01003.pdf>
7. CUDA C++ Programming Guide Memory Hierarchy, <https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html#memory-hierarchy>
8. CUDA C++ Programming Guide, <https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html>
9. R. Li, Y. Saad, GPU-accelerated preconditioned iterative linear solvers. *J. Supercomput.* **63**, 443–466 (2013). <https://doi.org/10.1007/s11227-012-0825-3>
10. S. Che, M. Boyer, J. Meng, D. Tarjan, J.W. Sheaffer, S.H. Lee, K. Skadron, Rodinia: a benchmark suite for heterogeneous computing, in *IEEE International Symposium on Workload Characterization (IISWC)*, Austin (2009)
11. Samples for CUDA Developers which demonstrates features in CUDA Toolkit, <https://github.com/NVIDIA/cuda-samples>
12. Intel FPGA SDK for OpenCL Pro Edition: Programming Guide, <https://www.intel.com/content/www/us/en/programmable/documentation/mwh1391807965224.html>
13. H.R. Zohouri, A. Podobas, S. Matsuoka, Combined spatial and temporal blocking for high-performance stencil computation on FPGA using opencl, in *Proceedings of the 2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, ser. FPGA'18* (ACM, New York, 2018), pp. 153–162. [Online]. Available: <https://doi.org/10.1145/3174243.3174248>
14. Introducing Innovations at 28 nm to Move Beyond Moore's Law, <https://www.intel.com.tw/content/dam/www/programmable/us/en/pdfs/literature/wp/wp-01125-stxv-28nm-innovation.pdf>
15. Khronos OpenCL Working Group, The OpenCL Specification: Version 1.0, 10 June 2009. [Online]. Available: <https://www.khronos.org/registry/OpenCL/specs/opencl-1.0.pdf>
16. H.R. Zohouri, N. Maruyama, A. Smith, M. Matsuda, S. Matsuoka, Evaluating and optimizing OpenCL kernels for high performance computing with FPGAs, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, Salt Lake City (2016)
17. Intel Corporation, Intel FPGA SDK for OpenCL: Best Practices Guide, 4 May 2018. [Online]. Available: https://www.altera.com/en_US/pdfs/literature/hb/opencl-sdk/aocl-best-practices-guide.pdf
18. Intel Corporation, Intel FPGA SDK for OpenCL: Programming Guide, 14 June 2018. [Online]. Available: https://www.altera.com/en_US/pdfs/literature/hb/opencl-sdk/aocl_programming_guide.pdf
19. D. Weller, F. Oboril, D. Lukarski, J. Becker, M. Tahoori, Energy efficient scientific computing on FPGA using OpenCL, in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, ser. FPGA'17* (ACM, New York, 2017), pp. 247–256. [Online]. Available: <https://doi.org/10.1145/3020078.3021730>
20. H. Esmailzadeh, E. Blem, R.S. Amant, K. Sankaralingam, D. Burger, Dark silicon and the end of multicore scaling, in *38th Annual International Symposium on Computer Architecture (ISCA)*, San Jose (2011)

Preliminary Performance and Programmability Comparison of the Thick Control Flow Architecture and Current Multicore CPUs



Martti Forsell, Sara Nikula, and Jussi Roivainen

1 Introduction

Multicore central processing units (CPU) are the workhorses of modern general purpose computing devices, such as computers, tablets, and smartphones. They were taken into commercial use over 15 years ago when it became evident that the clock speeds of single-core CPUs could not be increased any more. This was due to power density issues but at the same time integrating more transistors per chip seemed still possible [1, 2]. The main idea of multicore CPUs is to integrate multiple processor cores on a single chip and use them concurrently so that a P -core chip would execute the same functionality P times faster than a single-core processor. A precondition for this is that the functionality needs to be implementable as a parallel program. Multicore CPUs improve the performance over single-core processors for independent parallel tasks nearly linearly as long as the memory bandwidth is sufficient. Speedup is, however, difficult to find when dense intercommunication between the cores is required. Figure 1 shows the execution time of two C/pthreads programs—**matmul** and **matsum**—in an 18-core Intel Skylake Xeon W CPU as a function of number of threads (and processor cores) utilized in execution. Curves illustrating ideal scaling behavior are also shown for comparison purposes. In the case of **matmul**, the performance scales relatively well as the number of concurrent threads increases—the speedup 9.23 is achieved with 18 threads. On the contrary, as the number of parallel threads for **matsum** increases, the performance actually decreases by a factor of 4.72. This is caused by a higher degree of intercore memory traffic in **matsum** although **matmul** features more complex memory access pattern. In addition to the performance issues, the productivity of software development or

M. Forsell (✉) · S. Nikula · J. Roivainen
VTT, Oulu, Finland

e-mail: Martti.Forsell@VTT.Fi; Sara.Nikula@VTT.Fi; Jussi.Roivainen@VTT.Fi

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_36

493

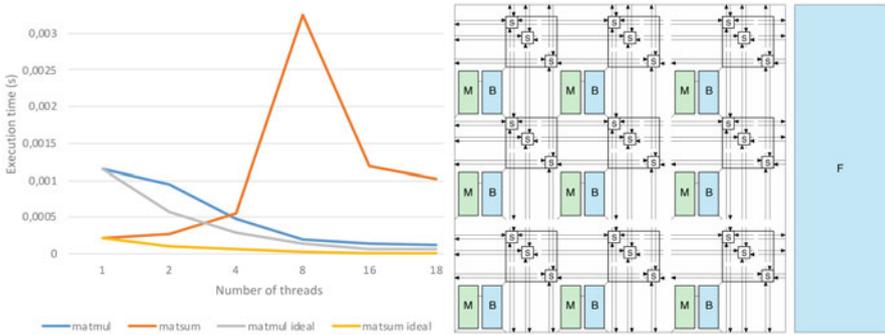


Fig. 1 Left: The execution time of the **matmul** and **matsum** test programs in an 18-core Intel Skylake Xeon W CPU as a function of the number of parallel threads. Additional curves illustrating ideal linear speedup behavior are shown. Right: Block diagram of TPA (F, frontend processing unit system; B, backend processing unit; M, memory module (or first-level cache); s, intercommunication switch). Spreading/return channel networks between FEs and BEs are not shown

programmability of multicore CPUs is far from perfect. Namely for performance reasons, programmers need to avoid straight-forward parallel processing patterns and replace them with more complex and error-prone structures as will be confirmed by our experiments. In parallel programs of Fig. 1 this can be seen as increase of the number of active code lines with respect to their textbook counterparts [3].

Our analysis indicates that the performance and programmability problems of multicore CPUs are caused by the weaknesses of current architectures rather than inefficient use of the methodology [4]. Particularly, the reasons for these problems include high costs of synchronizing and switching between threads, weakly scalable latency tolerance mechanisms and lack of support for key patterns of parallel computation. In order to solve these problems, we have introduced the *Thick Control Flow* (TCF) concept and outlined the *Thick Control Flow Processor Architecture* (TPA) for executing programs employing TCFs natively on our REPLICa multiprocessor framework [5–7]. A TCF is an abstraction of parallel computation that merges self-similar threads (called *fibers*) flowing through the same control path into one computational entity independently of the number of threads. The fibers within a TCF are executed *synchronously* with respect to each other to simplify parallel programming. While there are already a number of performance comparisons between TPA and its predecessors showing its potential [6, 8–10], it is not known how well TPA performs against current commercial multicore processors.

In this paper, we compare quantitatively the performance and programmability of TPA and Intel Skylake client and server multicore CPUs with a number of kernel programs that are useful in general purpose computing and in AI/ML. Code examples and qualitative observations on the goodness of the included programming approaches are given.

The rest of this paper consists of Sect. 2 describing the hardware architectures of the compared processors, Sect. 3 explaining shortly the programming methodologies used for producing the test programs, Sect. 4 performing the actual comparison with discussion on results, and Sect. 5 giving our conclusions.

2 Hardware Architectures

Hardware architectures of our interest targeted for general purpose parallel computing include TCF architectures [6], ESM architectures [11–14], their development versions and current commercial multicore architectures from Intel, Apple, AMD and IBM. In this preliminary performance comparison, we consider TPA, Intel Skylake client Core i7, and Skylake server Xeon W that will be utilized in the comparison of Sect. 4.

2.1 TPA

The *Thick Control Flow Processor Architecture* (TPA) is a scalable multiprocessor architecture that can be configured at design time for various constellations [6]. It belongs to our REPLICa multiprocessor framework that is aimed for addressing the performance and programmability issues of current general purpose multicore architecture [7]. TPA combines the TCF concept with the *emulated shared memory* (ESM) scheme. In ESM, the latency of the memory system is hidden via multi-threading and sufficient bandwidth, the synchronization cost is virtually eliminated using wave synchronization, and low-level parallelism exploitation is improved by chaining of functional units [11–13].

A TPA multiprocessor consists of F *frontend* (FE) processing units and B *backend* (BE) processing units (see Fig. 1). FEs take care of fetching instructions from the memory and executing the common parts of TCFs, such as control of the flow and base address computation. In turn, BEs handle execution of individual fibers. The memory system consists of two parts: FEs are connected to a traditionally organized SMP/NUMA memory system and BEs are attached to an ESM system employing a multimesh interconnect. The latter supports synchronous operations and access patterns such as concurrent reads and writes, reductions and multiprefix computations, as well as powerful compute-update operations. The FE and BE parts of the memory system are also connected together.

Execution happens by assigning the TCF in execution turn to a processor FE commanding a number of backend units. The FE fetches the instruction pointed by PC for execution and processes the common parts of the TCF in the FE's functional units. The FE passes BE operations to all assigned BEs for processing the individual fibers connected to the shared memory. The BEs get the common data/operands from the FE and execute operations assigned to fibers in the BE's functional units. A

step of execution is completed as soon as all the fibers have been once in execution. The architecture and methodology of TPA is reviewed in details in our white paper [7].

2.2 Core i7

Intel Core i7 6820HQ is a 64-bit 2.7 GHz (3.2 GHz turbo boost with all cores on) 4-core Skylake client microarchitecture CPU [15] (see Fig. 2). Its memory system features a 32 KB level 1 instruction and data caches per core, a 256 KB level 2 cache per core, a 2 MB shared level 3 cache per core, single memory controller, and 2 memory channels with maximum memory bandwidth of 31.79 MB/s. The microchip includes also a ninth-generation Intel HD Graphics 530. The processor was tested on an Apple MacBook Pro (late 2016) laptop computer with 16 GB of onboard 2133 MHz LPDDR3 SDRAM.

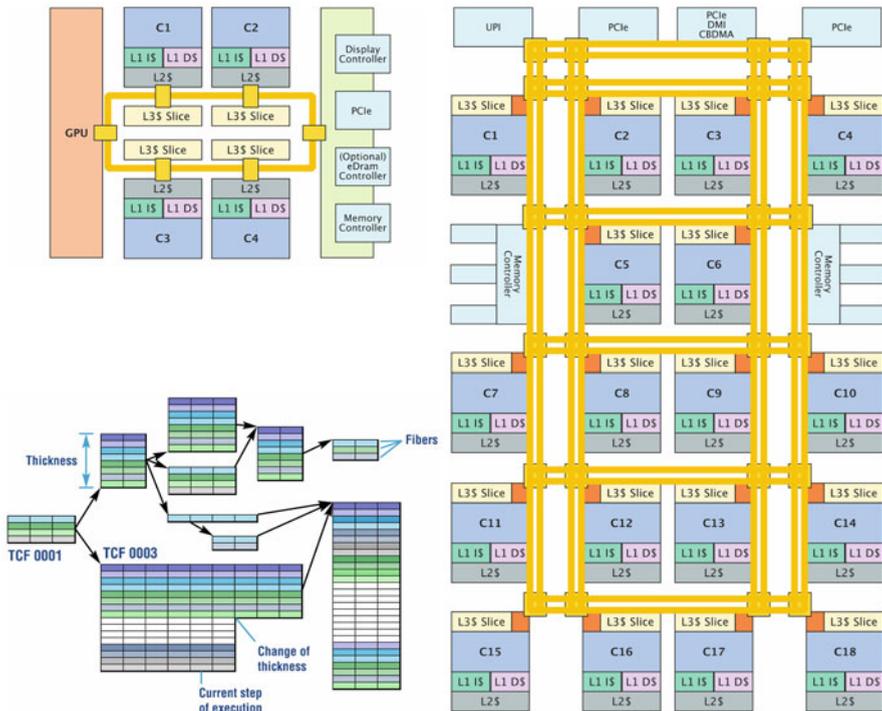


Fig. 2 Top left: Block diagram of a 4-core Intel Skylake client Core i7 (Cn = processor core n). Bottom left: A program consisting of 10 TCFs while the fifth instruction of TCF 0003 of thickness 16 is being executed. Right: Block diagram of a 18-core Intel Skylake server Xeon W

2.3 Xeon W

Intel Xeon W-2191B is a 64-bit 2.3 GHz (3.2 GHz turbo boost with all cores on), 18-core Skylake server microarchitecture CPU [16] (see Fig. 2). Its memory system features 32 KB level 1 instruction and data caches per core, 1 MB level 2 cache per core, 1.375 MB level 3 cache per core, and 2 memory controllers and 4 memory channels with maximum memory bandwidth of 79.47 MB/s. The processor was tested on an Apple iMac Pro (late 2017) workstation with 128 GB of onboard 2666 MHz EEC DDR4 SDRAM.

3 Programming Methodologies

The main idea of parallel computation is to decompose or divide the computational problem at hands into subproblems that can be solved in parallel and to compose the solution of the original problem from the results of the subprograms. This may naturally happen hierarchically, recursively, and/or in consequent parts. Solving the subproblems in parallel introduces the need to communicate between the parallel parts, which in turn creates dependencies that require *synchronization* between the parallel parts. Finally, to get practical results, the parallel parts need to be executed in physical processors, which raises a need to define the relationship of execution units and parallel solutions known as *mapping*. A processor can be said to have good programmability if the functionalities can be expressed compactly and naturally without unnecessary architecture-dependent constructs and migration between different implementations executing efficiently in different hardware configurations is as simple as possible.

There are huge number of parallel programming models/languages taking different approaches. In this section, we focus on TCF programming scheme and pthreads that will be utilized in TPA and Skylake CPUs, respectively.

3.1 Thick Control Flows

The Thick Control Flow (TCF) concept is an abstraction of parallel computation that merges self-similar threads (called fibers) flowing through the same control path into computational entities (called TCFs) independently of the number of threads [5]. The number of fibers is called thickness. A programmer can dynamically change the thickness of a TCF during execution. The fibers within a TCF are executed synchronously with respect to each other to enable simple parallel programmability. This kind of a concept shares many properties of idealized parallel random access machine (PRAM) model of computation [17]. A PRAM consists of a number of processors running under a single clock connected to a synchronous shared memory

with idealized latency properties. There exists a well-developed theory of parallel algorithms for PRAM [3, 12].

TCFs have a single control, but they can process multiple data elements in parallel. When a TCF with thickness T calls a subroutine, the subroutine is not called separately by each of the T fibers, but the TCF calls it only once with thickness T . A call stack is not related to each fiber but to each of the TCFs, since fibers do not have program counters. This implies that stack variables many times have thickness T reflection the fact that there is a data element for each fiber. Multiple TCFs can be executed in parallel in multiple FEs. Figure 2 shows an example of a program consisting of 10 TCFs. The TCF 0003 is in execution and features a change of thickness from 16 to 8 after one more step.

In a TCF system, such as our TPA processor, the fibers of the currently executed TCF are evenly distributed to the backend execution units. The execution units operate in parallel and maintain synchronicity between consecutive instructions so that the TCF concept executes as the programmer expects.

3.2 *POSIX Threads*

POSIX threads (pthreads) is a set of C language interfaces (functions, header files) for threaded programming. It allows a program to control multiple different threads of computational work that overlap in time [18]. Each thread executes its instructions independently. All threads in a process share the memory space, functions, data, and files. In addition, thread-specific data can also be defined. Any thread can create new threads and threads can have different kinds of relations with each other, such as master–slave or producer–consumer. Synchronization is needed to ensure that multiple threads are collaborating correctly to avoid problems like deadlocks and race conditions. Synchronization can be done via mutual exclusion locks, semaphores, join functions, and barriers, which can be implemented by a set of functions provided. Threads in different processes can be synchronized via synchronization variables in the shared memory [18].

In a typical symmetric multiprocessing (SMP) system, such as Apple MacBook Pro and Apple iMac Pro running Apple Mac OS operating system utilized in this paper, threads are periodically assigned to processor cores with least amount of work. Therefore, a parallel program written in pthreads is often actually executed in parallel in SMP systems.

4 Comparison

In order to compare the goodness of TPA versus Skylake multicore processors, we performed a series of quantitative performance and programmability tests, as well

as collected qualitative programmability observations. In addition, program code examples are given.

4.1 Quantitative Measurements

We measured the execution time and counted the number of active code lines for seven short program kernels (see Table 1) that represent widely used functionalities of parallel computing on three multiprocessor systems representing TPA and Intel Skylake CPUs introduced in Sect. 2 (see Table 2). Each program except **bprefix** and **sync** was implemented as a single TCF version for TPA and three pthreads versions for Skylake CPUs—straight-forward version, matched parallelism version, and blocked versions. The *straight-forward* pthreads versions are written similarly as the TCF versions except that synchronizations are added to guarantee correct execution order of operations assigned to different cores. The matched parallelism versions limit the number of threads to the given maximum, which in our case is the number of processor cores P . The matching is done by employing loops that process at most P operations at the time. We expect that matched parallel versions are substantially faster than straight-forward ones. This is because matching eliminates interference between time slots defined by the operating system scheduler and actual computation, as well as the process management overhead, especially in the case of fine-grained parallel functionality. The blocked versions divide the processed data elements to blocks that are processed in the processor cores in parallel. This

Table 1 Benchmark programs

block	Copies an array of 1024/262144 64-bit integers to another location in the memory (Tests memory access)
lprefix	Calculates the prefix sum of an array of 1024/65536 64-bit integers using the logarithmic prefix sum algorithm (Tests demanding memory access pattern and change of thickness, blocking version)
bprefix	Calculates the prefix sum of an array of 262144 64-bit integers using the blocking prefix sum algorithm (Tests blocking-style prefix computation—includes only one version since there is no straight-forward variant of this other than lprefix)
madd	Adds an array of 1024/262144 64-bit integers to another similar array (Tests exclusive memory access and basic matrix operation)
mmul	Multiplies two arrays of 128x128 64-bit integers to a third similar array (Tests exclusive and concurrent memory access and basic matrix operation—includes only one version due to built-in LLVM optimizations)
threshold	Applies a threshold filter to an array of 1024/262144 64-bit integers (Tests compute-update operations)
sync	Performs a barrier synchronization of threads (Tests synchronization cost)

Table 2 Tested multiprocessors

	Core i7	Xeon W	TPA-16
Number of cores	4	18	1 FE / 16BE
Clock (sustained/peak all cores/peak 1 core)	2.7/3.2 GHz	2.3/3.2/4.2 GHz	3.2 GHz
Number of FUs (FE/BE)	8	8	4 / 10
Number of Mus	3	3	1
Threading scheme	HT	HT	TCF

FE frontend, *BE* backend, *MU* memory unit, *HT* Intel's 2-way simultaneous hyperthreading, *TCF* thick control flow scheme

should also improve the performance over the matched parallelism versions due to increased locality and reduced interprocessor communication.

The TPA configuration (TPA-16) used in the comparison was selected so that: (i) it represent a typical entry-level constellation and (ii) its integer-only silicon area (estimated to be 19 mm² at an artificial 11 nm process [8]) does not exceed that of the Skylake client without its on-chip GPU (50.354 mm² at Intel's 14 nm process). The Skylake server chip (485 mm² at Intel's 14 nm process) is included for comparison purposes to get an idea how Skylake designs scale to high-end versions and how that performs with respect to TPA. In addition to silicon area, the number of functional units in each processor core is roughly the same but the Skylake processor cores feature three memory units while there is only one unit per BE in TPA.

The multicore CPU computers were running Apple MacOS Mojave 10.14.6. The test programs were compiled with the Apple LLVM compiler with the -O3 optimization on expect for **lprefix** matched parallelism and blocking versions that were compiled with the -O2 optimization. To simplify testing setup and to avoid additional SIMD-specific optimizations, we selected not to use AVX units in the Skylake systems. Each run executed the programs repeatedly up to million times so that slow startup time, effects of dynamic voltage frequency scaling (DVFS) and cooling were virtually eliminated and the frequency was kept as close to the frequency of 3.2 GHz. Besides having multiple iterations for each run, we ran each test five times and use the average time of the runs as the measurement result for the comparison. Since our TPA configurations do not include the outermost part of the memory system, we selected the problem sizes included in the comparison so that the data set fits to the L2 caches of the Skylake CPUs.

The TPA programs were executed in our in-house TPASim simulation software that provides a clock/RTL-accurate model of the TPA configuration used in the tests. There was no need to make multiple runs with TPA since in the lack of DVFS and outer memory system effects, the simulator always gives the same results.

The straight-forward version results of our measurements are shown as relative performance and relative length of the active program code lines in top row of Fig. 3. From these, we can make the following observations:

- TPA-16 provides vastly higher performance than the Skylake CPUs. It executed the straight-forward test programs that perform actual computation (**block**,

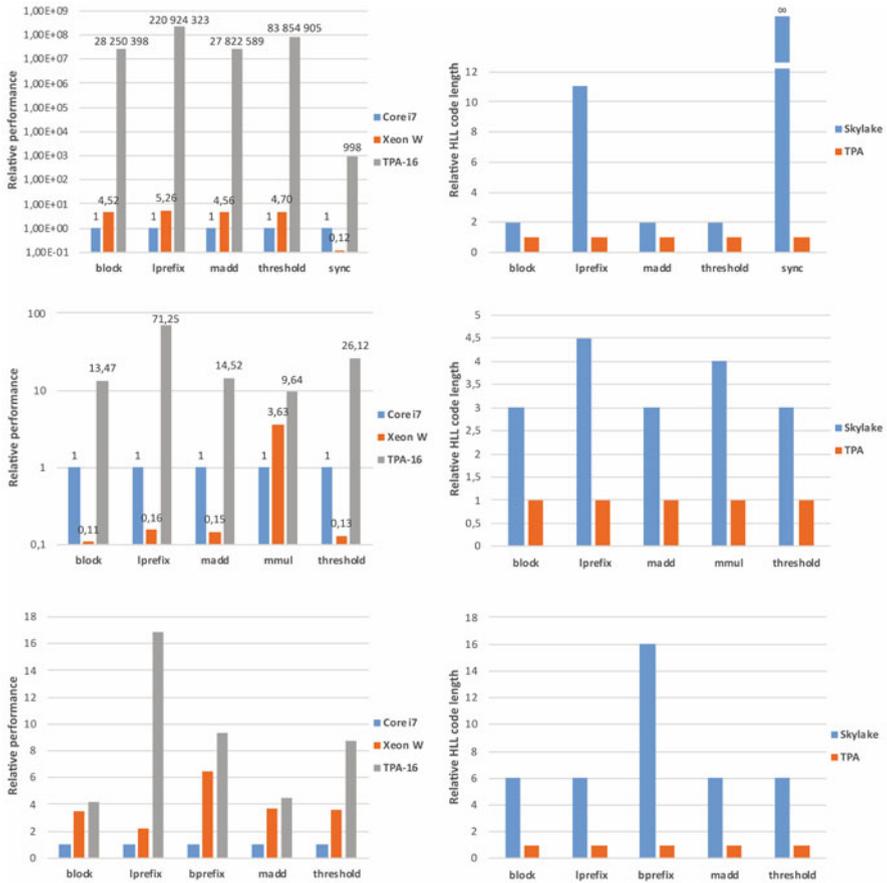


Fig. 3 Relative performance of the straight-forward, matched parallelism and blocking test programs in TPA-16 and Intel Skylake multicore CPUs (left column). Relative source code length of the straight-forward, matched parallelism and blocking test programs for TPA and Intel Skylake multicore CPUs (right column). Note that the scale is logarithmic for the upper performance charts

lprefix, madd, threshold) 61.8 million times faster than Core i7. The boost of switching Core i7 to Xeon W is 4.75-fold, reducing the average speedup of TPA-16 over Xeon W to still very high 13.0 million.

- The barrier synchronization program **sync** executes in TPA-16998 times faster than in Core i7 and 8154 times faster than in Xeon W. Core i7 executes this faster since there are 4.5 times fewer processor cores to synchronize.
- The Skylake test programs are longer than those for TPA. The number of active code lines in the test programs in Skylake processors doing actual computation (**block, lprefix, madd, threshold**) is 3.06 times higher than that in TPA.
- Synchronization does not need any extra effort in TPA due to synchronous nature of TCFs, while in Skylake CPUs, one needs to perform synchronizations

explicitly. As a result, the relative code length of a synchronization is infinitely shorter in TPA.

The matched parallelism version results are shown as relative performance and relative length of the active program code lines in middle row of Fig. 3:

- The average performance of TPA-16 in executing matched parallel versions of the test programs is 20.37 times higher than that of Core i7 and 78.30 times higher than that in Xeon W.
- Despite of its larger number of processor cores, Xeon W performs 3.84 times worse than Core i7 in these tests in average due to the higher intercommunication network latency and risk of congestion. TPA-16 does not share this problem although its network [19] is almost as large as in Xeon W.
- The **mmul** test program performs differently that other programs since the LLVM compiler recognized it and replaced the written baseline algorithm with a more optimized one.
- With respect to TPA the active code line count overhead in Skylake jumps from at least 2 to at least 3. This is caused by addition of looping to match the software parallelism with the hardware one in Skylake.

The blocked version measurement results are shown as relative performance and relative length of the active program code lines in bottom row of Fig. 3:

- Introduction of blocking decreases the average relative performance advantage of TPA-16 versus Core i7 down to 7.61. The highest speedup is achieved with the **lprefix** test program due to inter-thread dependencies and change of thickness.
- Xeon W performs now close to TPA-16 in the **block** and **madd** test programs, while the average performance advantage of TPA-16 versus Xeon W drops down to 2.07.
- The blocked Skylake versions increases the code line count overhead with respect to TPA from at least 3 to at least 6. In average, the blocked test programs were 7.3 times longer in Skylake than in TPA.

The TPA test programs implemented for this study are very compact. We compared them to sequential programs solving the same computational problems. It turned out that TPA versions are even shorter than their sequential counterparts (see Fig. 4). This is because loops for going through data elements of arrays are not needed if TCFs are used for programming.

4.2 Programming Examples

In order to illustrate the practical programming differences between TPA and Skylake CPUs, let us have a look at the **lprefix** test program that computes the prefix sum of an array **A** of N integers using the logarithmic algorithm. It is the most complex test program utilizing constantly altering intercore communication

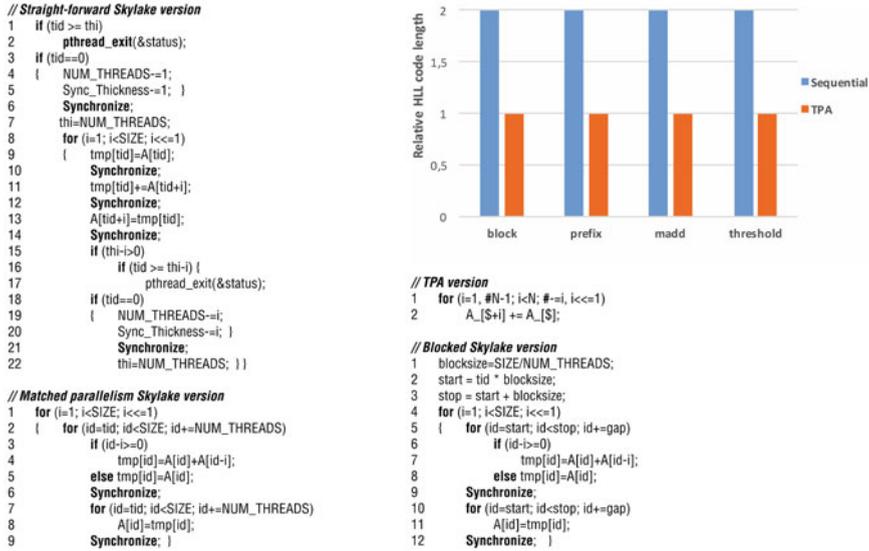


Fig. 4 The high-level language versions of the lprefix test program for TPA and Skylake. (# = thickness of the TCF, \$ = fiber identifier, tid = thread identifier, NUM_THREADS = number of threads in pthreads system, thi = thread private number of threads, Sync_Thickness = thread count information needed by barrier synchronization, SIZE = N = problem size). Note that for simplicity, the matched parallelism and blocked versions are not using exactly the same algorithm as the TCF and straight-forward versions. Top right: Relative source code length of the sequential and TPA test programs

pattern and change of thickness during the execution. Figure 4 shows the TPA version of lprefix written in C-style parallel TCF language as well as straight-forward, matched parallelism and blocked Skylake versions in C/pthreads. These are the same program versions as used in the tests of Sect. 4.1.

The TPA version consists just of two active code lines and corresponds to a typical parallel computing textbook version of the algorithm [3, 12]. The functionality is implemented with a single TCF that executes the loop while its thickness decreases iteratively by exponentially increasing steps starting from one.

The straight-forward Skylake version does the same thing as the TPA versions, but needs explicit synchronizations due to asynchronous execution of threads in Skylake. For the same reason, a temporary variable tmp is needed since the actual data array A changes its content along with the execution introducing risk of reading old data. Explicit pthreads_exit() functions are needed to adjust the number of threads along the execution. As a result, the number of active code lines is 11 times higher than in TPA.

The straight-forward Skylake versions have catastrophically poor performance. The programmer may therefore want to limit the number of threads. The matched parallelism Skylake version drops the number of parallel threads from N down to 4 in Core i7 and 18 in Xeon W. This happens by turning the parallel statements into

loops that process just 4 or 18 elements in parallel. We dropped the altering thickness scheme of the algorithm since it would have been quite complex to decrease the number of threads with the loops so that the resulting behavior matches exactly to the straight-forward version. As a result, the number of active code lines drops to 9. If the altering thickness scheme would have been included the number of code lines would have been higher than in the straight-forward version.

The matched parallelism versions have also unnecessary low performance, especially for Xeon W. This problem can be partially be solved by minimizing the intercore communication. The blocked Skylake test programs partition the data to blocks of N/P elements, where P is the number of cores, to maximize the locality of references and to avoid inter-core traffic as much as possible. The blocking increases the number of active program lines to 12.

4.3 Qualitative Observations

We collected also qualitative observations on programmability and performance-related aspects of the Skylake and TPA systems from an expert programmer with a lot of experience and a person not familiar with the theories of parallel computing nor either of the used programming methodologies. The set of programs the expert and inexperienced programmer created included the programs used in our comparison. The beginner's observations related to programming include:

- Pthreads feels an understandable way to describe parallel implementation but in the case of unexpected results, that happen way too often, the reason behind the problems is hard to discover and solve.
- Programming TPA feels in principle simpler than that for Skylake, but to be successful TPA definitely needs better tools than current academic quality ones.
- Trying out TPA assembler in parallel program implementation seemed quite impossible at the first glance but turned out easier than expected. There was no need to try out X86 assembler for Skylake but getting a parallel program working that way would have been difficult due to low-level complexity of pthreads.
- Sometimes synchronizing threads in Skylake programs was more troublesome than writing the program itself.

The expert observations include notes on higher-level concepts behind the actual tools for programming and the resulting performance:

- Getting decent performance out of parallel functionality targeted for Skylake multicore CPUs is much more difficult than that for TPA. The main issues with the Skylake are asynchronous nature of thread execution, high cost of synchronizations, threading model that is in practice bounded above to the number of HW threads.
- The factors that make parallel programming easy for TPA are the simplicity of getting the right amount of parallelism with the TCF model, synchrony of exe-

cution that makes scheduling the operations to smoothly advancing consecutive steps easy, and insensitivity to partitioning and mapping choices that makes code portable.

- Even though TPA uses a quite complex VLIW instruction set with chaining, writing programs in assembler makes some sense for critical inner loops.
- The performance portability and scalability are quite hard to achieve with Skylake CPUs, while in TPA it is easy since TPA's programming concept directs a programmer automatically to simple and adaptive code.

Based on these qualitative observations, parallel programmability of TPA is in way better shape than that of Skylake multicore CPUs even though they have very strong set of available tools, as well as wide support of a large processor manufacturer, the software community, and schools teaching programming and parallelism. One remarkable indicator of the weakness of the current approach is that the paradigm of software engineering is still sequential although multiprocessors have been exclusively used for more than 15 years.

5 Conclusions

We have made a preliminary performance and programmability comparison of a TCF architecture TPA against Intel Skylake client and server multicore CPUs Core i7 and Xeon W. According to our measurements, TPA performs much better than Intel Skylake Core i7. The straight-forward parallel implementations of test programs, representing widely used primitives of parallel computing, give 61.8 million times better performance with a half of active code lines in TPA. If the parallelism of programs is matched to that of Skylake hardware, the TPA performance advantage is still 20.37-fold while the Skylake code line count overhead increases to three. By maximizing the locality using partitioning of data and functionality to blocks, the Skylake is able to drop the overhead to 7.61 but cost of this is at least six times longer programs. We compared TPA also to high-end Intel Skylake Xeon W processor with more cores than in TPA. The performance results were mostly a couple of times more favorable to Skylake, but the program complexity overhead remains as high as that for the smaller Skylake CPU while the estimated silicon area overhead is 10-fold. The qualitative observations from our programmers indicate that TPA is way simpler to program than a Skylake CPU. Likewise, getting expected performance for parallel functionalities is easier with TPA.

Our future work plans include comparing the performance and programmability of multicore CPUs thoroughly to different TPA configurations. For that, additional benchmarks, programming languages, and other processors will be involved.

Acknowledgment This work was funded by VTT and the grant 319759 of Academy of Finland.

References

1. International Technology Roadmap for Semiconductors (ITRS), year 2003 edition, Semiconductor Industry Association (SIA), <http://www.itrs.net>
2. Research at intel from a few cores to many: A Tera-scale computing research overview, white paper, Intel (2006)
3. J. Jaja, *Introduction to Parallel Algorithms* (Addison-Wesley, Reading, 1992)
4. M. Forsell, Accelerating general purpose parallel computing with the TPA architecture, ScalPerf 18, September 23–28, 2018, Bertinoro, Italy
5. V. Leppänen, M. Forsell, J.-M. Mäkelä, Thick control flows: Introduction and prospects, Proc. PDPTA'11, July 18–21, 2011, Las Vegas, USA, 540–546
6. M. Forsell, J. Roivainen, V. Leppänen, Outline of a thick control flow architecture, Proc. MPP'16/SBAC-PAD'16, October 26–28, 2016, Marina del Rey Marriott, Los Angeles, USA
7. REPLICA multiprocessor framework, white paper, VTT (2020)
8. M. Forsell, J. Roivainen, V. Leppänen, J. Träff, Implementation of multioperations in thick control flow processors, Proc. APDCM'18, May 21–25, 2018, Vancouver, Canada
9. M. Forsell, Flexible fibering scheme for thick control flow processors, Proc. PDPTA'18, July 30–August 2, 2018, Las Vegas, USA, 16–20
10. M. Forsell, J. Roivainen, V. Leppänen, J. Träff, Supporting concurrent memory access in TCF processor architectures. *Microprocess. Microsyst.* **63**, 226–236 (2018)
11. A. Ranade, How to emulate shared memory. *J. Computer Sys. Sci.* **42**, 307–326 (1991)
12. J. Keller, C. Keßler, J. Träff, *Practical PRAM programming* (Wiley, New York, 2001)
13. M. Forsell, A scalable high-performance computing solution for network on chips, *IEEE Micro* **22**, 5 (September–October 2002), 46–55
14. U. Vishkin, Using simple abstraction to reinvent computing for parallelism. *Commun. ACM* **54**(1), 75–85 (2011)
15. Skylake (client) - Microarchitectures - Intel, [https://en.wikichip.org/wiki/intel/microarchitectures/skylake_\(client\)](https://en.wikichip.org/wiki/intel/microarchitectures/skylake_(client)). Accessed 21 Mar 2020
16. Skylake (server) - Microarchitectures - Intel, [https://en.wikichip.org/wiki/intel/microarchitectures/skylake_\(server\)](https://en.wikichip.org/wiki/intel/microarchitectures/skylake_(server)). Accessed 21 Mar 2020
17. S. Fortune, J. Wyllie, Parallelism in random access machines, proc. STOC'78, San Diego, California, USA — May 1–3, 1978, 114–118
18. B. Lewis, D. Berg, *PThreads primer: A guide to multithreaded programming*, Sunsoft Press (1996)
19. M. Forsell, J. Roivainen, V. Leppänen, The REPLICA on-chip network, NORCAS'16, November 1–2, 2016, Copenhagen, Denmark

Part VI
**Communication Strategies, Internet
Computing, Cloud, and Computational
Science**

Refactor Business Process Models with Redundancy Elimination



Fei Dai, Huihui Xue, Zhenping Qiang, Lianyong Qi, Mohammad R. Khosravi, and Zhihong Liang

1 Introduction

Business processes are the key artifacts of process-aware information systems [1], thus enterprises must be able to deal with their quality issues [2, 3]. Similar to other conceptual models, business process models should be easily understandable. Therefore, understandability is regarded as one of the most important among quality criteria [4, 5]. Today, with the board use of PAIS, thousands of business process models [6, 7] are modeled for a variety of purposes, for example, process analysis and process enactment. It has been observed that when these models are required to adapt to real-world demands, introducing model redundancies is inevitable.

Against this background, a question arises here is how to improve the understandability of business process models. Related literature shows that the size of the model is a significant influence [8].

Refactoring business process models refers to the process of modifying the internal structure of business process models without changing the external behavior [9]. Although there are several refactoring techniques for business process models

F. Dai · H. Xue · Z. Qiang (✉) · Z. Liang

School of Big Data and Intelligent Engineering, Southwest Forestry University, Kunming, China
e-mail: daifei@swfu.edu.cn; zhliang@swfu.edu.cn

L. Qi

School of Information Science and Engineering, Qufu Normal University, Jinning, China
e-mail: lianyongqi@qfnu.edu.cn

M. R. Khosravi

Department of Computer Engineering, Persian Gulf University, Bushehr, Iran

Department of Electrical and Electronic Engineering, Shiraz University of Technology, Shiraz, Iran

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_37

509

through behavior-preserving model transformations [10], these techniques cannot be used to eliminate elements in models.

To solve this problem, we first propose a process model smell for identifying redundant elements in business process models, where the metric of this smell is an implicit place. For these IPs, we remove them from these business process models to decrease the size of these models, that is, the decrease of the number of nodes makes these models easier to understand such that the understandability of these models can be improved.

The contributions of this study are as follows:

1. We are the first to propose a process model smell for identifying redundant elements in business process models using unfolding, where the metric of this smell is an implicit place (IP).
2. We are the first to present an algorithm for computing IPs from the complete finite prefix unfolding (CFPU) rather than the reachability graph (RG) of a net system.
3. We implemented our approach and experiments show our approach can find IPs from business process models efficiently and eliminate them without altering their models' behavior.

The remainder of this paper is organized as follows. Section 2 introduces a motivating example. Section 3 presents the definitions used throughout this paper. Section 4 presents a process model smell. Section 5 proposes three refactoring operations. Section 6 describes the implementation and evaluation of our approach. Section 7 discusses related work. Section 8 concludes this paper.

2 Motivation

Figure 1 shows a business process model from [11] describes the service process of a train station, where the circles denote places and the rectangles denote tasks. First, a customer sends a request message (transition *cus_ts_request*) to the train station and the train station checks the ticket availability by interacting with the component availability through the transitions *ts_ava_info*, *ava_ts_infoAvail*, and *ava_ts_itinerary*. After checking, the train station communicates with the booking component through the book message (*ts_booking_book*) and the *ack* message (*booking_ts_ack*). Finally, when the component booking sends the invoice message (*booking_cus_invoice*) to the customer or executes an invisible action (transition *tau*), the train station sends back the result message (*ts_cus_result*) to the customer.

The places p_1 , p_3 , and p_6 circled by dash line in Fig. 1 are redundant. This is because the place p_3 results in an unreasonable synchronization and the places p_3 and p_6 are unuseful input places for the transitions *ts_ava_info* and *ts_booking_book*.

Based on the above analysis, these redundant places should be removed from Fig. 1. Figure 2 shows the refactored business process model. Intuitively, the size of the

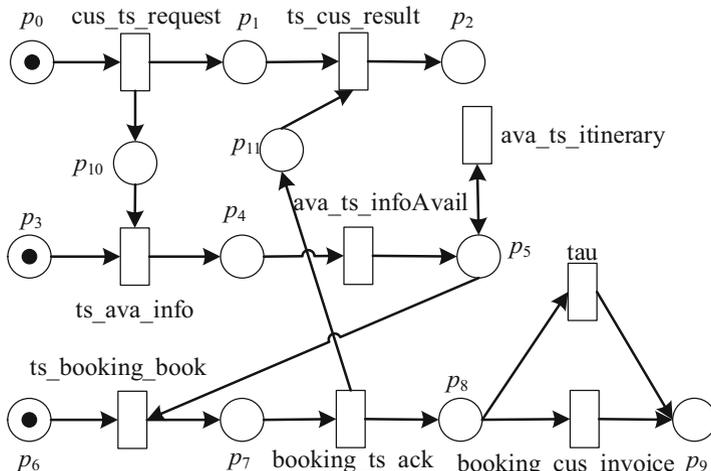


Fig. 1 A refactored business process model with more efficiency

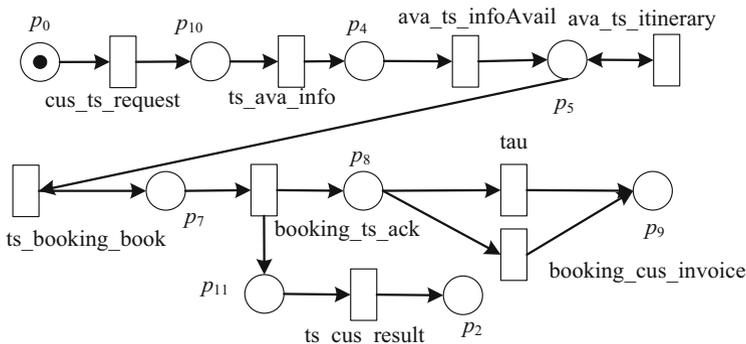


Fig. 2 A refactored business process model

refactored model is less than that of the original model. Besides, from the behavior perspective, the refactored business process model is trace-equivalent to the original model. So this refactoring operation can decrease the size of the model such that the understanding of this model can be improved.

3 Preliminaries

Definition 1 (Petri net) A Petri net is a triple $N = (P, T, F)$, where:

1. P is a set of places.
2. T is a set of transitions.

3. $F \subseteq (P \times T) \cup (T \times P)$ is a flow relation.

State M is the marking of N and can be defined as: $M: P \rightarrow \{0, 1, 2, 3, \dots\}$. M_0 often refers to the initial state of N .

Denote $X = (P \cup T)$, for a node $x \in X$, $\bullet x = \{y \in X \mid (y, x) \in F\}$ is called the preset of x ; $x^\bullet = \{y \in X \mid (x, y) \in F\}$ is called the postset of x .

Definition 2 (Petri net Semantics) [12] Let $N = (P, T, F)$ be a Petri net.

1. $M: P \rightarrow \mathbb{N}$ is a marking of \mathbb{N} i, where \mathbb{N} i is the set of non-negative integer. A marking is the distribution of tokens over places. M denotes all states of N . $M(p)$ denotes the number of tokens in place p .
2. For $\forall t \in T$, t is said to be enabled in M , if $\forall p \in \bullet t: M(p) \geq 1$;
3. If an enabled transition t in M_1 fires, the marking evolves from M_1 to M_2 such that $M_2 = M_1 - \bullet t + t^\bullet$. That is, the firing of t leads from M_1 to M_2 , denoted by $M_1[t > M_2$.
4. A net system is a pair $\Sigma = (N, M_0)$, where N is a net and M_0 is the initial state of N .

If there is a firing transition sequence $\sigma \in T^*$ that leads from marking M_1 to M_2 , this can be denoted by $M_1[\sigma > M_2$. $[M_1 >$ denotes the set of markings reachable from M_1 .

Definition 3 (Labeled Transition System) A labeled transition system is a tuple $LTS = (S, s_0, A, \Delta)$, where:

1. S is a set of states.
2. $s_0 \in S$ is the initial state.
3. A is a set of labels.
4. $\Delta \subseteq S \times A \times S$ is a transition relation.

The reachability graph $RG(N, M_0)$ of a Petri net N with M_0 is the labeled transition system, where $[M_0 >$ is the set of states, M_0 is the initial state, T is the set of labels, and the transition relations are defined as $\{(M_0, t, M') \mid M_0, M' \in [M_0 > \wedge M_0[t > M'\}$.

Definition 4 (Implicit Place) [13] A net system $\Sigma = (N, M_0)$, a place $p \in P$ is called implicit iff $\forall t \in p^\bullet, \forall M \in RG(N, M_0): M \geq \bullet t \wedge \{p\} \Rightarrow M \geq t^\bullet$, that is, the removal of p preserves all the firing sequences of Σ .

The definition of an occurrence net is based on the following three concepts [14].

1. If there are two nodes x and y on a path of an occurrence net, x and y are in *causal* relation, denoted by $x < y$.
2. If two paths are leading to x and y , which start at the same place and immediately diverge in an occurrence net, x and y are in *conflict* relation, denoted by $x \# y$.
3. If x and y are neither $x < y$ nor $y < x$ nor $x \# y$, x and y are in concurrency relation, denoted by x *co* y .

Definition 5 (Occurrence Net) [15] An occurrence net system $O = (B, E, G)$ such that:

1. $\forall b \in B: |\bullet b| \leq 1$.
2. O is acyclic, or, equivalently, the causal relation is a partial order.
3. $\forall x \in B \cup E$: it holds $\neg(x \# x)$ and $\{y \in B \cup E \mid y < x\}$ is finite.
4. $Min(O)$ denotes the set of minimal elements of $B \cup E$ with respect to the causal relation.

B is called the set of conditions, and E is called the set of events. We can see that the relation between every two elements of an occurrence net is unique, that is, one relation in $<$, $\#$, and co .

Definition 6 (Branching Process) [15] A branching process of a net system $\Sigma = (P, T, F, M_0)$ is a tuple $\pi = (O, h) = (B, E, G, h)$, where h satisfies the following properties:

1. $h(B) \subseteq P$ and $h(E) \subseteq T$.
2. For every $e \in E$, the restriction of h to $\bullet e$ is a bijection between $\bullet e$ and $\bullet h(e)$, and similarly for e^\bullet and $h(e)^\bullet$.
3. The restriction of h to $Min(O)$ is a bijection between $Min(O)$ and M_0 (π “starts” at M_0).
4. For every $e_1, e_2 \in E$, if $\bullet e_1 = \bullet e_2$ and $h(e_1) = h(e_2)$ then $e_1 = e_2$.

The maximal branching process of a net system is called the *unfolding* of a net system [14, 16].

Definition 7 (Complete Finite Prefix) [17] Let $\Sigma = (N, M_0)$ be a net system and $\pi = (O, h)$ a branching process with $N = (P, T, F)$ and $O = (B, E, G)$, where:

1. $\forall e \in E, [e] = \{x \in E \mid x < e \vee x = e\}$ is a local configuration. For an event $e \in E$, the *local configuration without itself* $! [e]$ is defined as $\{x \in E \mid x < e \vee x \neq e\}$.
2. $X \subseteq B$ is called *co-set*, iff $\forall c_1, c_2 \in B$ implies $c_1 co c_2$.
3. For a finite configuration $C \subseteq E$, the co-set $Cut(C) = (Min(O) \cup C) \setminus \bullet C$ is a *cut*. $Mark(C) = h(Cut(C))$ is the reachable marking by firing the transitions whose corresponding events are in C .
4. An adequate order $<$ is a partial order on local configurations such that for two events $e, f \in E, [e] \subset [f]$ implies $[e] < [f]$.
5. $\forall e \in E, e$ is a *cut-off event* if there exists another event $e' \in E$ such that $Mark([e]) = Mark([e'])$ and $[e'] < [e]$.
6. A complete finite prefix unfolding (CFPU) is the greatest backward closed subnet of a branching process containing no events after any cut-off event.

Example 1 Figure 3a, b show a Petri net and its related RG, respectively. Σ_1 has four concurrent transitions (transitions t_4, t_5, t_6 , and t_{12}), while its reachability graph has 86 states. We can see that the number of states in the reachability graph grows exponentially with the number of concurrent transitions. Since constructing reachability graphs of Petri nets may encounter the state space explosion problem, it

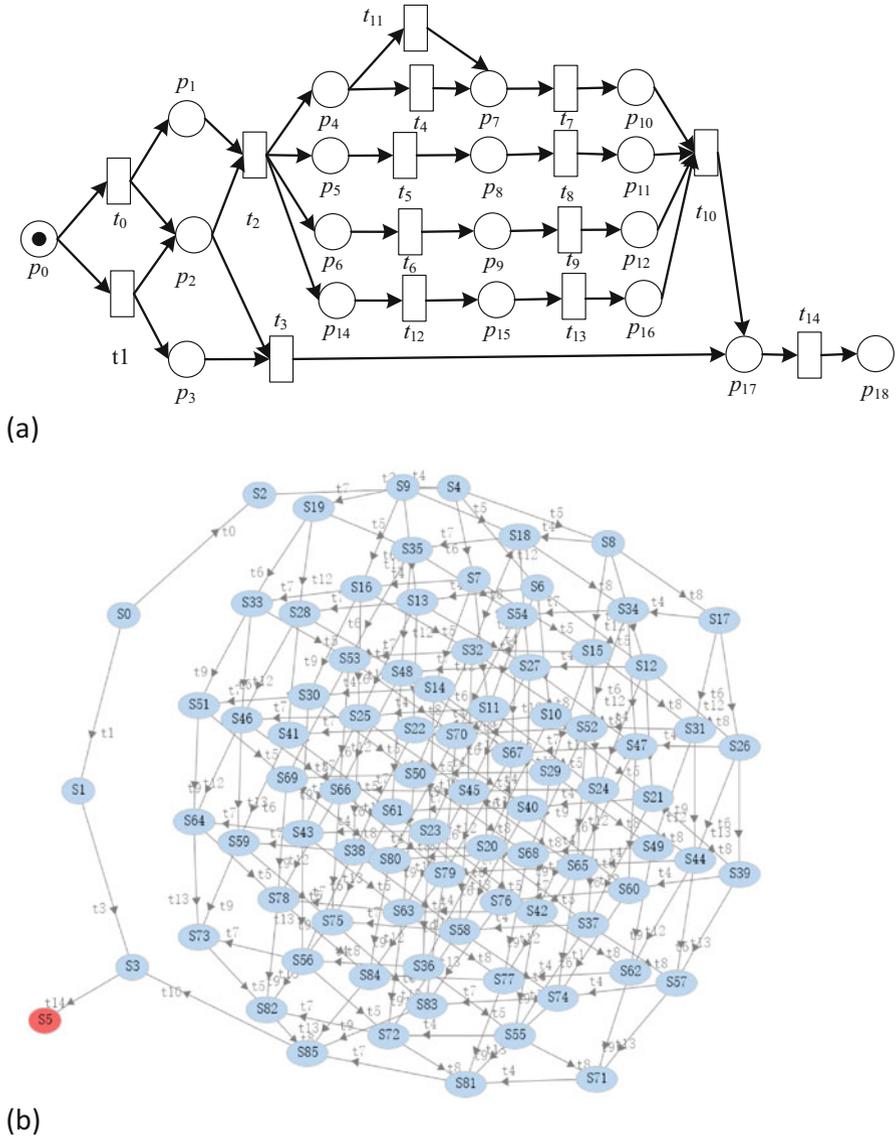


Fig. 3 A net system Σ_1 and its related RG. (a) A Petri net Σ_1 . (b) The reachability graph of Σ_1

is hard to compute IPs based on the reachability graph. Therefore, we will propose an efficient algorithm for computing IPs in Sect. 4.

4 Identify a Process Model Smell

Code smells mean bad code fragments in the field of software engineering, which are used to identify refactoring opportunities [18]. Similarly, we propose a process model smell for identifying IPs in business process models using the unfolding technique.

4.1 A Process Model Smell

We present a process model smell that is used to assist model designers in identifying redundant elements in business process models. In the following, this process model smell is briefly described and then detected by checking an implicit place. Finally, this smell can be addressed by the behavior-preserving refactoring operations in Sect. 5.

Description Elements in a business process that result in unreasonable synchronizations or unuseful inputs are redundant. These redundant elements should be removed to improve the understandability of a process model.

Metric The metric of this process model smell is an implicit place. If there is an implicit place in a business process model, the model has a redundant element, which can be removed.

4.2 Computation of IPs from the CFPU

In this section, we analyze the relation between IPs in the reachability graph and their manifestation in the CFPU.

In Fig. 4a, the place p_3 (circled by dash line) is an implicit place, so we have $M_3 = \{p_3, p_5\} \geq \bullet t_5 \setminus \{p_3\} \Rightarrow M_3 = \{p_3, p_5\} \geq \bullet t_5$ in Fig. 4b. In its CFPU shown Fig. 4c, we can see that the place p_3 's corresponding conditions are $p_3 - 4$ and $p_3 - 6$, that is, $h(p_3 - 4) = h(p_3 - 6) = p_3$, where $Cut([t_2 - 2]) \geq \bullet t_5 - 4 \setminus \{p_3 - 4\} \Rightarrow Cut([t_2 - 2]) \geq \bullet t_5 - 4$. This is a manifestation of the place p_3 in the CFPU.

Note that we label each event and condition with its corresponding transition and place label, respectively, “-”, and its construction order in the CFPU. For example, $t_1 - 1$ is an event that is a first-generated event in the CFPU, and t_0 is the corresponding transition in the original net. From $t_1 - 1$, we have $(h(t_1 - 1) = t_1)$. In Fig. 4c, we can also have $[t_4 - 5] = \{t_1 - 1, t_3 - 3\}$ and $Mark([t_4 - 5]) = \{p_3, p_4\}$.

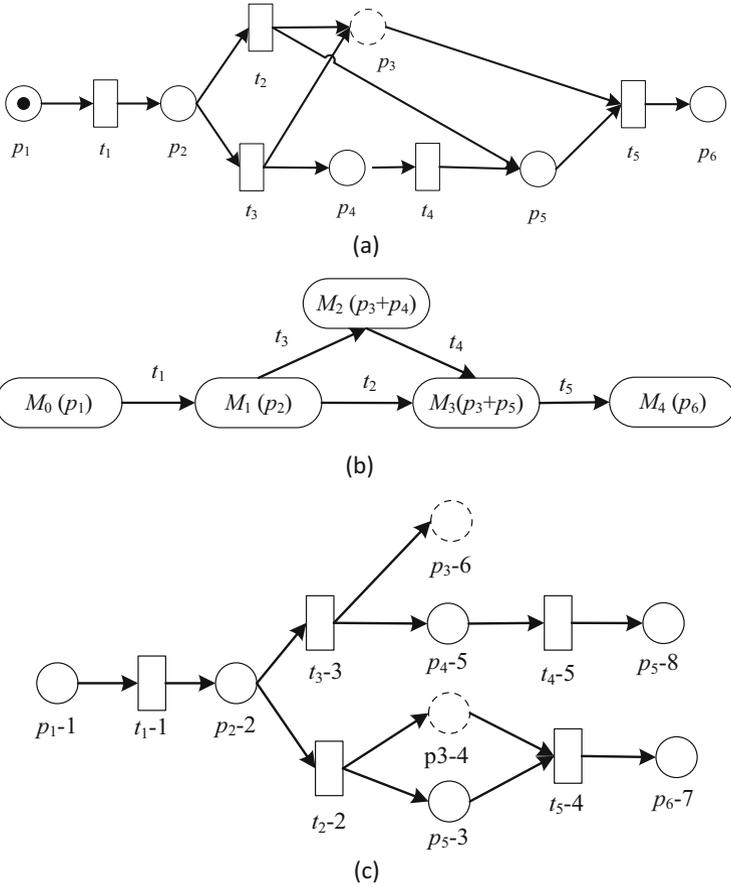


Fig. 4 A net system Σ_2 , its related RG, and its related CFP. (a) An implicit place p_3 in a Petri net Σ_2 . (b) The reachability graph of Σ_2 . (c) A manifestation of the implicit place p_3 in CFP of Σ_2

With the analysis above, we define the corresponding condition of an IP in the CFPU below.

Definition 9 (Implicit Condition) Let $\pi = (O, h)$, $O = (B, E, G)$ be a CFPU of a net system $\Sigma = (N, M_0)$. $b \in B$ is an implicit condition (IC), iff $\forall e \in b^* : Cut(![e]) \geq *e \setminus \{b\} \Rightarrow Cut(![e]) \geq *e$.

An implicit condition is that whose removal does not change all the local configurations of π .

Based on Definition 9, the equivalence between an IP in the RG and its corresponding IC in the CFPU can be defined below.

Proposition 1 Let $\pi = (O, h)$, $O = (B, E, G)$ be a CFPU of a net system $\Sigma = (N, M_0)$, $N = (P, T, F)$. Let $p \in P$ is an implicit place, if and only if $\exists B' \subseteq B: \forall b \in B'$ is an implicit condition, and $h(b) = p$.

Proof (\Rightarrow) Since $p \in P$ is an implicit place, for each transition t in p^\bullet and each reachable marking M in $R(N, M_0)$, if $M \geq \bullet t \setminus \{p\}$, $M \geq \bullet t$. From the properties of the complete finite prefix, we know that:

1. From $p \in P$ and $\forall t \in p^\bullet$, there exists a condition set B' and an event e of π such that $\forall b \in B'$, $h(b) = p$, $h(e) = t$ (h preserves the nature of nodes).
2. From $\forall M \in R(N, M_0)$ and $M \geq \bullet t$, there exist an event e and a cut c of π such that $h(e) = t$, $h(c) = M$, and that the event e is enabled at c , that is, $c \geq \bullet e$. For the cut c , it holds $Cut(![e]) = c$. Since $Cut(![e]) = c$ and $c \geq \bullet e$, we have $Cut(![e]) \geq \bullet e$.
3. From $\forall t \in p^\bullet$, $\forall M \in R(N, M_0): M \geq \bullet t \setminus \{p\} \Rightarrow M \geq \bullet t$, there exist a condition set B' , an event e , and its $![e]$ of π such that for each condition in B' and each event e in b^\bullet , $Cut(![e]) \geq \bullet e \setminus \{b\} \Rightarrow Cut(![e]) \geq \bullet e$, $h(b) = p$, $h(e) = t$, $h(Cut(![e])) = M$.

Therefore, it holds that $\forall b \in B'$ is an implicit condition if $\forall e \in b^\bullet: Cut(![e]) \geq \bullet e \setminus \{b\} \Rightarrow Cut(![e]) \geq \bullet e$ and $h(b) = p$. Thus, we have proved that $\exists B' \subseteq B: \forall b \in B'$ is an implicit condition, and $h(b) = p$.

(\Leftarrow) By the definition of the implicit condition, if b is an implicit condition of π , we have $\forall e \in b^\bullet: Cut(![e]) \geq \bullet e \setminus \{b\} \Rightarrow Cut(![e]) \geq \bullet e$. By the isomorphism between π and Σ , we know that there exists a place p and a reaching marking M of Σ such that $h(b) = p$, $h(e) = t$, $h(Cut(![e])) = M$ and that $\forall t \in p^\bullet: M \geq \bullet t \setminus \{p\} \Rightarrow M \geq \bullet t$. From $\forall t \in p^\bullet: M \geq \bullet t \setminus \{p\} \Rightarrow M \geq \bullet t$. From the principle of locality for Petri nets, we can deduce that $\forall t \in p^\bullet$, $\forall M \in R(N, M_0): M \geq \bullet t \setminus \{p\} \Rightarrow M \geq \bullet t$. On the other hand, by applying the properties of the complete finite prefix, we know that there exists a condition set B' of π such that $\forall b \in B': h(b) = p$. Thus, we have for $\forall t \in p^\bullet$, $\forall M \in R(N, M_0): M \geq \bullet t \setminus \{p\} \Rightarrow M \geq \bullet t$ and therefore p is an implicit place.

4.3 An Algorithm for Computing IPs from the CFPU

We propose an efficient algorithm for computing IPs from the CFPU based on Esparz's unfolding algorithm [14]. There are three steps:

4.3.1 Compute the Local Configuration Without Itself of Some Event

Given an event e , we present an algorithm for computing its local configuration without itself denoted by $![e]$, which can be found in Algorithm 1.

In Algorithm 1, the branching process denoted by BP begins to construct the corresponding conditions of the initial marking of Σ (Line 1). $PE(BP)$ denotes the set of possible extensions of BP (Line 2). $h(x)$ denotes the corresponding transition

of the event x . New events from the possible extensions Ext are added into BP at a time with their output conditions (Line 5).

Definition 10 (Possible Extension) [14] Let $\Sigma = (N, M_0)$ be a net system and $\pi = (O, h)$, a branching process with $N = (P, T, F)$ and $O = (B, E, G)$, and $t \in T$ is a transition with output places p_1, \dots, p_n . An event $e = (t, X)$ is a possible extension of $\pi \{n_1, \dots, n_k\}$ satisfying the following condition:

1. $X = \{b_1, b_2, \dots, b_n\}$ is a co-set conditions of π .
2. $h(X) = {}^*t$.
3. $\{n_1, \dots, n_k, e, (p_1, t), \dots, (p_n, t)\}$ is also a branching process.

Algorithm 1 compute the local configuration without itself of some event

```

input: a net system  $\Sigma = (N, M_0)$ , and an event  $x$ 
output:  $![x]$ 
1  $BP \leftarrow \{(b_1, \phi), \dots, (b_n, \phi)\}$ , where  $h(\{b_1, \dots, b_n\}) = M_0$ 
2  $Ext \leftarrow PE(BP)$ 
3 while  $Ext \neq \emptyset$  do
4   if  $t \neq h(x)$  then
5     add to  $BP$  an event  $e = (t, X)$  of  $Ext$  and a condition  $(p, e)$ 
   for every output place  $p$  of  $t$ ;
6      $Ext \leftarrow PE(BP)$ ;
7      $![x] \leftarrow ![x] \cup \{t\}$ ;
8   if  $t = \phi(x)$  then
9     return  $![x]$ 

```

4.3.2 Compute an Implicit Condition

We present an algorithm for checking whether a condition is implicit, which can be found in Algorithm 2. First, the preset of the condition is computed (Lines 1–3). Secondly, each event e 's $![e]$ is computed using Algorithm 1 (Lines 4–5). Finally, each event of the preset is checked whether it is satisfying Definition 9 (Lines 6–8).

Algorithm 2 compute an implicit condition

```

input: a net system  $\Sigma = (N, M_0)$ , a CFPU  $\pi = (B, E, G, h)$ ,
and a condition  $b$ 
output: true or false
1 foreach  $e \in E$  do
2   if  $e \in b^*$  then
3      $E' \leftarrow E' \cup \{e\}$ 
4 foreach  $e \in E'$  do
5    $![e] \leftarrow$  call Algorithm 1( $\Sigma, e$ )
6   if  $!(Cut(![e]) \geq^* e \setminus \{b\} \Rightarrow Cut(![e]) \geq^* e)$  then
7     return false
8 return true

```

4.3.3 Compute an Implicit Place

We present an algorithm for checking whether a place is implicit, which can be found in Algorithm 3. First, the CFPU of a net system Σ is computed (Line 1). Secondly, the place p corresponding conditions of the CFPU is computed (Lines 2–4). Finally, each corresponding condition is checked whether it is implicit (Lines 5–8).

Algorithm 3 compute an implicit place

```

input: a net system  $\Sigma=(N, M_0)$  and a place  $p$ 
output: true or false
1  $\pi=(B, E, G, h) \leftarrow$  call Esparz's unfolding algorithm
2   foreach  $b \in B$  do
3     if  $h(b) == p$  then
4        $B' \leftarrow B \cup \{b\}$ 
5   foreach  $b \in B'$  do
6     if ! call Algorithm2( $\Sigma, \pi, b$ ) then
7       return false
8   return true

```

5 Refactoring Operations

We propose refactoring operations that are used to cope with the discussed process model smell in Sect. 4. These refactoring operations can remove IPs from a business process model without changing the model's behavior. Specifically, there are three types of refactoring operations:

1. For the implicit place without any input transition, we delete this place and the related arc. That is, this place is an unuseful input place.
2. For the implicit place with one input transition and one output transition, we delete this place and the related arcs. That is, this place causes an unreasonable synchronization.
3. For the implicit place without any output transition, we delete this place and the related arc. That is, this place is an unuseful output place.

Algorithm 4 eliminate an implicit place

```

input: a net system  $\Sigma=(N, M_0)$  and a place  $p$ 
output:  $\Sigma'=(N', M'_0)$ 
1 if  $p = \phi$  then
2    $M_0 \leftarrow M_0 - \{p\}; P = P - \{p\}$ 
3   foreach  $t \in p$  do
4      $F \leftarrow F - \{(p, t)\}$ 
5 if  $p' = \phi$  then

```

```

6    $P \leftarrow P - \{p\}$ 
7   foreach  $t \in \mathring{p}$  do
8      $F \leftarrow F - \{(t, p)\}$ 
9   if  $|p^*| = 1 \wedge |p| = 1$  then
10     $P \leftarrow P - \{p\}$ 
11    foreach  $t \in \mathring{p}$  do
11      $F \leftarrow F - \{(t, p)\}$ 
12    foreach  $t \in p^*$  do
13      $F \leftarrow F - \{(p, t)\}$ 
14     $P' \leftarrow P; T' \leftarrow T; F' \leftarrow F; M'_0 \leftarrow M_0;$ 
15    return  $\Sigma'$ 

```

The algorithm for eliminating an implicit place from a business process model can be found in Algorithm 4.

To prove that our proposed refactoring operations do not change models' behavior, we first introduce the trace notion.

Given a net system $\Sigma = (N, M_0)$, the trace is the set of firing sequences of transitions, denoted by $Trace(N, M_0)$. Based on the trace notion, the reachability graph $RG(N, M_0)$ can also be defined below.

$$RG(N, M_0) = \{M_0[\sigma > M' \mid \sigma \in Trace(N, M_0)\}.$$

Proposition 2 Let $\Sigma = (N, M_0)$ be a net system. If $\Sigma' = (N', M'_0)$ is obtained by eliminating an implicit place from Σ , then $Trace(N, M_0) = Trace(N', M'_0)$.

Proof By the definition of an implicit place, we have $RG(N, M_0) = RG(N', M'_0)$. From $RG(N, M_0) = \{M_0[\sigma > M' \mid \sigma \in Trace(N, M_0)\}$ and $RG(N', M'_0) = \{M'_0[\sigma' > M' \mid \sigma' \in Trace(N', M'_0)\}$, we can deduce that $Trace(N, M_0) = Trace(N', M'_0)$. Therefore, we have $Trace(N, M_0) = Trace(N', M'_0)$.

6 Experiments

We implemented a prototype tool based on an open-source tool of Petri nets named PIPE (The Platform Independent Petri net Editor) [19] to automate our approach. The screenshot of the prototype tool is shown in Fig. 5.

To evaluate the effectiveness and efficiency of our approach, we conducted experiments on a Windows 10 machine running on a PC with 2.50 GHz i5 CPU and 8GB of RAM, running Windows 10. Our database includes 40 examples, where 30 examples are taken from the literature and 10 examples are hand-crafted.

First, to evaluate the effectiveness of our approach, we need to show that our approach can decrease the size of business process models with redundant elements and our proposed refactoring operations do not change models' behavior.

Table 1 shows some experimental results. For each case, the table gives the size of places ($|P|$), transitions ($|T|$) and flows ($|F|$) in the original model, the size of places ($|P'|$), transitions ($|T'|$) and arcs ($|F'|$) in the refactored model, and the behavioral comparison of the original model and the refactored model (Equivalence). During

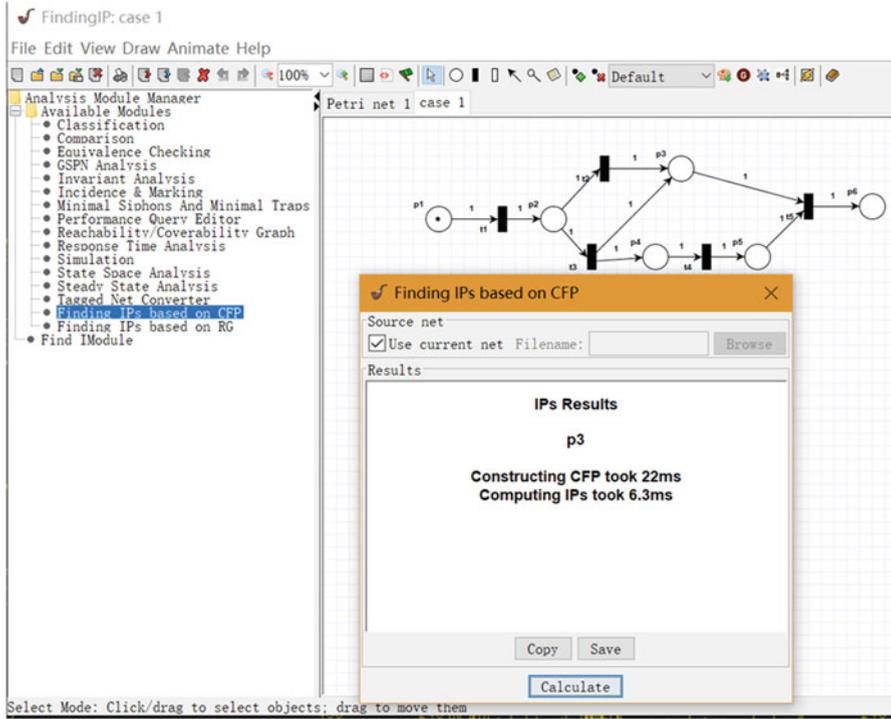


Fig. 5 The screenshot of our implementation

Table 1 Experimental results

id	The original model Σ			The refactored model Σ'			Equivalence
	$ P $	$ T $	$ F $	$ P' $	$ T' $	$ F' $	
case 1	8	5	7	7	5	12	$Trace(\Sigma) = Trace(\Sigma')$
case 2	6	5	13	5	5	10	$Trace(\Sigma) = Trace(\Sigma')$
case 3	12	6	24	8	6	16	$Trace(\Sigma) = Trace(\Sigma')$
case 4	6	4	12	5	4	8	$Trace(\Sigma) = Trace(\Sigma')$
case 5	6	6	12	5	6	10	$Trace(\Sigma) = Trace(\Sigma')$
case 6	9	7	22	8	7	19	$Trace(\Sigma) = Trace(\Sigma')$
case 7	17	13	35	16	13	32	$Trace(\Sigma) = Trace(\Sigma')$
case 8	9	6	19	7	6	13	$Trace(\Sigma) = Trace(\Sigma')$
case 9	20	15	40	19	15	38	$Trace(\Sigma) = Trace(\Sigma')$
case 10	12	9	22	9	9	18	$Trace(\Sigma) = Trace(\Sigma')$

the experiments, we can see that $|P'|$ and $|F'|$ in the refactored model are less than ($|P|$) and ($|F|$) in the original model, respectively, and that these two models are trace-equivalent. That is, our approach can decrease the size of the original models and preserve the behavior of these models.

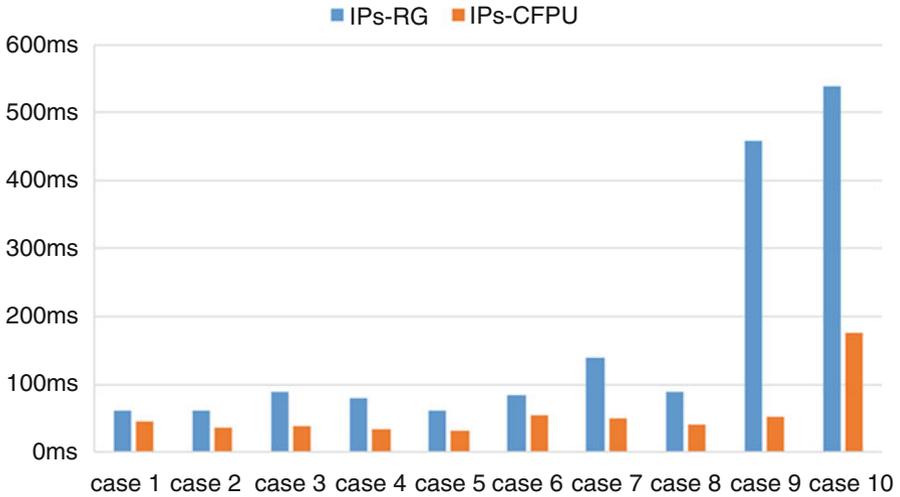


Fig. 6 Time comparison of IPs-CFPU and IPs-RG

Second, to evaluate the efficiency of our approach, we need to show that computing IPs from the CFPU (IPs-CFPU) takes less time than computing IPs from the RG (IPs-RG). Figure 6 shows that our IPs-CFPU approach has good performance. When there is more concurrency in the original models (case 9 and case 10), our approach has superior performance. For example, our approach takes 52.2 milliseconds, while the IPs-RG approach takes 459 milliseconds in case 9. The time of our approach is reduced by 88.6% compared with the IPs-RG approach.

7 Related Work

There have been many works on the understandability of business process models.

Size, separability, density, token split, connectivity, and connector degree are regarded as important model factors that influence the understandability of business process models [8, 20]. In [21], the authors conducted an empirical study on process model understanding and its impact factors. The experiment results showed that density and average connector degrees are very important factors. In [22], the authors undertook an empirical evaluation to show that high cross-connectivity makes models to be easier understood. The authors in [8] found that both personal and model factors affect model understanding where model factors include nodes, arcs, tasks, connectors, control flow complexity, depth, etc.

To improve the understandability of activity tags in a business process model, Leopold et al. proposed an approach based on the corpus's second-order similarity in [23] to automatically annotate activity tags in the business process model. Further,

Mendling et al. studied the effect of the activity tag style on the comprehensibility for models in business process modeling practice in [21]. Leopold et al. proposed an automatic refactoring technique for converting the activity tags of the action-noun style into the verb-object style in [24]. These works mentioned above mainly focuses on improving the understandability of the activity tags in the business process model.

However, these refactoring techniques above cannot be used to eliminate redundant elements in business process models.

To the best of our knowledge, the similar work is in [9]. The authors in [9] regarded the process fragments that contain the same flow control logic to be redundant because these fragments may cause inconsistency if one change in one is made. The proposed refactoring operation is to abstract these redundant fragments as a complex activity and then to reference this activity in the original model. Our work is different from this work in two aspects. First, our approach regards IPs as redundant fragments in a business process model rather than the process fragments with the same flow control logic. Second, our proposed refactoring operations are to remove redundant fragments from business process models, while the refactoring operation proposed in [9] is to replace redundant process fragments by reference.

8 Conclusions

In this paper, we propose an approach to refactor business process models with redundancy elimination. First, the process model smell is proposed to identify redundant elements in a business process model using unfolding, where the metric of this smell is an IP. Second, three refactoring operations are proposed to remove IPs from the business process model and preserve the behavior of this model. Finally, experimental results show the efficiency and effectiveness of our approach.

Acknowledgments This work was supported in part by the Project of National Natural Science Foundation of China under Grant No. 61702442, 61862065, and 61662085, the Application Basic Research Project in Yunnan Province Grant No. 2018FB105.

References

1. L.J. Wen, J.M. Wang, W.M.P. van der Aalst, B.Q. Huang, Mining process models with prime invisible tasks. *Data Knowl. Eng.* **69**(10), 999–1021 (2010)
2. Q. Mo, W. Song, F. Dai, L. Lin, T. Li, Development of collaborative business processes: A correctness enforcement approach, *IEEE Trans. Serv. Comput.* (2019), <https://doi.org/10.1109/TSC.2019.2961346>
3. X. Xu, X. Liu, Z. Xu, F. Dai, X. Zhang, L. Qi, “Trust-oriented IoT service placement for smart cities in edge computing,” *IEEE Internet Things J.* (2019), <https://doi.org/10.1109/JIOT.2019.2959124>.

4. R. Laue, A. Gadatsch, A. “measuring the understandability of business process models - are we asking the right questions?”, in *Business Process Management Workshops. BPM 2010. Lecture Notes in Business Information Processing*, ed. by M. zur Muehlen, J. Su, vol. 66, (Springer, Berlin, Heidelberg, 2010), pp. 37–48
5. X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, W. Dou, Dynamic Resource Provisioning with Fault Tolerance for Data-Intensive Meteorological Workflows in Cloud, *IEEE Trans. Ind. Inf.* (2019), <https://doi.org/10.1109/TII.2019.2959258>
6. H. Leopold, J. Mendling, H.A. Reijers, M.L. Rosa, Simplifying process model abstraction: Techniques for generating model names. *Inf. Syst.* **39**, 134–151 (2014)
7. X. Xu, W. Dou, X. Zhang, J. Chen, EnReal: An energy-aware resource allocation method for scientific workflow executions in cloud environment. *IEEE Trans. Cloud Comput.* **4**(2), 166–179 (2016)
8. H. A. Reijers and J. Mendling, “A Study Into the factors that influence the understandability of business process models, *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.*, 2011, vol. 41, no. 3, pp. 449–462
9. R. Dijkman, B. Gfeller, J. Küster, H. Völzer, Identifying refactoring opportunities in process model repositories. *Inf. Softw. Technol.* **53**(9), 937–948 (2011)
10. B. Weber, M. Reichert, J. Mendling, H.A. Reijers, Refactoring large process model repositories. *Comput. Ind.* **62**(5), 467–486 (2011)
11. G. Salaün, T. Bultan, N. Roohi, Realizability of choreographies using process algebra encodings. *IEEE Trans. Serv. Comput.* **5**(3), 90–302 (2012)
12. T. Jin, J. Wang, Y. Yang, L. Wen, K. Li, Refactor business process models with maximized parallelism. *IEEE Trans. Serv. Comput.* **9**(3), 456–468 (2016)
13. W.M.P. van Aalst, T. Basten, Inheritance of workflows: An approach to tackling problems related to change. *Theor. Comput. Sci.* **270**(1–2), 125–203 (2002)
14. J. Esparza, R. Stefan, V. Walter, An improvement of McMillan’s unfolding algorithm. *Formal Methods Syst. Des.* **20**(3), 285–310 (2002)
15. J.M. Colom, M. Silva, Improving the linearly based characterization of P/T nets, in *Advances in Petri nets 1990, LNCS*, vol. 483, (Springer Verlag, Berlin, Heidelberg, 1991), pp. 113–145
16. J. Pei, L. Wen, X. Ye, A. Kumar, Z. Lin, “Transition adjacency relation computation based on unfolding: Potentials and challenges,” [C]// In: *Proc of the OTM Confederated International Conferences “on the Move to Meaningful Internet Systems,” 2016*, 61–79
17. C. Girault, R. Valk. *Petri nets for systems engineering: A guide to modeling, verification, and applications[M]* (2003)
18. T. Mens, T. Tourwe, A survey of software refactoring. *IEEE Trans. Softw. Eng.* **30**(2), 126–139 (2004)
19. N.J. Dingle, W.J. Knottenbelt, T. Suto, PIPE2: A tool for the performance evaluation of generalised stochastic Petri nets. *Meas. Model. Comput. Syst.* **36**(4), 34–39 (2009)
20. J. Mendling, H.A. Reijers, J. Cardoso, What makes process models understandable? in *Proc. the 5th International Conference on Business Process Management*, (Springer (BPM 2007), Berlin, Heidelberg, 2007), pp. 48–63
21. J. Mendling, H.A. Reijers, J. Recker, Activity labeling in process modeling: Empirical insights and recommendations. *Inf. Syst.* **35**, 467–482 (2010)
22. I. Vanderfeesten, H.A. Reijers, J. Mendling, W.M.P. van der Aalst, J. Cardoso, On a quest for good process models: The cross-connectivity metric, in *Advanced Information Systems Engineering (CAiSE 2008).*, Lecture Notes in Computer Science, vol. 5074, (Springer, Berlin, Heidelberg, 2008), pp. 480–494
23. H. Leopold, C. Meilicke, M. Fellmann M, F. Pittke, H. Stuckenschmidt, J. Mendling, Towards the automated annotation of process models, in *Proc of the International Conference on Advanced Information Systems Engineering*. Springer International Publishing (2015), pp. 401–416
24. H. Leopold, S. Smirnov, J. Mendling, Refactoring of process model activity labels, in *Proc of the International Conference on Application of Natural Language to Information Systems*, (Springer, Berlin, Heidelberg, 2010), pp. 268–276

A Shortest-Path Routing Algorithm in Bicubes



Masaaki Okada and Keiichi Kaneko

1 Introduction

Recently, large-scale computation demands are increasing in many fields. However, each processor has a performance limitation. Because a single processor takes much time to perform large-scale computation, parallel computation, which aims to reduce computation time by distributing tasks to multiple processors, has been attracting much attention.

In a parallel system, the topology of its interconnection network, that is, the way to interconnect the multiple processing elements, has strong influence on the performance of the system. The topology of the interconnection network can be treated in the graph theory by replacing its processing elements with vertices and its links with edges.

Among the various topologies [1–8], proposed for interconnection networks, we focus on the bicube [4]. The bicube is a topology that is obtained as a variant of the hypercube [5], which has been widely used for various parallel systems. The bicube can connect the same number of the vertices as the hypercube by almost half links. Also, it has the good property of the vertex symmetry. Hence, it can easily implement larger-scale parallel systems. However, there have not been proposed any shortest-path routing algorithms in the bicube. Therefore, in this study, we aim to propose a shortest-path routing algorithm in bicubes.

The rest of this paper is organized as follows. The definitions and notations with respect to the shortest-path routing in a bicube are presented in Sect. 2. Next, comparison of the cube-based topologies for interconnection networks is conducted

M. Okada · K. Kaneko (✉)

Tokyo University of Agriculture and Technology, Tokyo, Japan
e-mail: s185080s@st.go.tuat.ac.jp; k1kaneko@cc.tuat.ac.jp

in Sect. 3. Then, in Sect. 4, we propose our shortest-path routing algorithm in a bicube. Finally, we conclude this paper in Sect. 5.

2 Definitions

In this section, we give the definitions of related topics and the properties of topologies.

Definition 1 An n -dimensional hypercube, Q_n , is an undirected graph whose vertex set is $\{0, 1\}^n$. For two vertices $\mathbf{u} = (u_1, u_2, \dots, u_n)$ and $\mathbf{v} = (v_1, v_2, \dots, v_n)$ in Q_n , they are adjacent if and only if $H(\mathbf{u}, \mathbf{v}) = 1$ where $H(\mathbf{u}, \mathbf{v})$ represents the Hamming distance between \mathbf{u} and \mathbf{v} . □

The number of vertices and the diameter of Q_n are 2^n and n , respectively. Q_n is a symmetric graph whose degree is n . Also, Q_n has a recursive structure such that it consists of two Q_{n-1} . Figure 1 shows an example of Q_4 .

Definition 2 For a bit sequence $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0) (\in \{0, 1\}^n)$, define a function $p(\mathbf{u})$ such that $p(\mathbf{u}) = 0$ if $\sum_{i=0}^{n-1} u_i$ is even and $p(\mathbf{u}) = 1$ if $\sum_{i=0}^{n-1} u_i$ is odd. Then, for two bit sequences $\mathbf{u}, \mathbf{v} \in \{0, 1\}^n$, we call \mathbf{u} and \mathbf{v} are in lp-relation if and only if $\mathbf{u} = \mathbf{v}$ and $p(\mathbf{u}) = p(\mathbf{v}) = 0$ or $\mathbf{u} = \bar{\mathbf{v}}$ and $p(\mathbf{u}) = p(\mathbf{v}) = 1$. □

For example, two bit sequences $(1, 0, 0, 1, 1, 0)$ and $(0, 1, 1, 0, 0, 1)$ are in lp-relation.

Definition 3 An n -dimensional bicube, B_n , ($n \geq 3$) is an undirected graph whose vertex set is $\{0, 1\}^n$. For a vertex $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ in B_n , it has n adjacent vertices $\mathbf{u}^{(i)}$ ($n - 1 \geq i \geq 0$) where $\mathbf{u}^{(i)}$ ($n - 2 \geq i \geq 0$) is given by $\mathbf{u}^{(i)} = (u_{n-1}, u_{n-2}, \dots, u_{i+1}, \bar{u}_i, u_{i-1}, \dots, u_0)$ and $\mathbf{u}^{(n-1)}$ is given depending on the parity of n . That is, if n is odd, $\mathbf{u}^{(n-1)} = (\bar{u}_{n-1}, v_{n-2}, v_{n-3}, \dots, v_0)$, where $(v_{n-2}, v_{n-3}, \dots, v_0)$ is the bit sequence that is in lp-relation with $(u_{n-2}, u_{n-3}, \dots, u_0)$. The edge $(\mathbf{u}, \mathbf{u}^{(n-1)})$ is called a complementary edge. If n is even, $\mathbf{u}^{(n-1)} = (\bar{u}_{n-1}, u_{n-2}, v_{n-3}, \dots, v_0)$ where $(v_{n-3}, v_{n-4}, \dots, v_0)$ is the bit sequence that is in lp-relation with $(u_{n-3}, u_{n-4}, \dots, u_0)$. □

Figure 2 shows an example of B_4 . The number of vertices and the diameter of B_n are 2^n and $\lceil (n + 1)/2 \rceil$ ($n \geq 7$), respectively. On the other hand, the diameters

Fig. 1 Example of Q_4

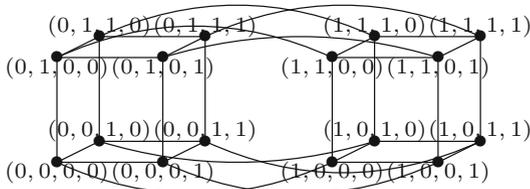
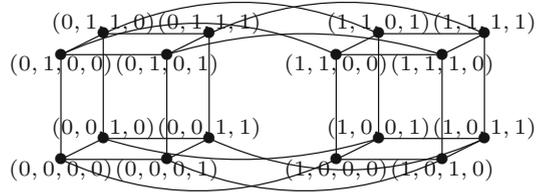


Fig. 2 Example of B_4



of B_3 , B_4 , B_5 , and B_6 are 3, 4, 4, and 5, respectively. B_n is a vertex-symmetric graph whose degree is n . Moreover, the subgraphs induced by the vertex sets $\{(u_{n-1}, u_{n-2}, \dots, u_0) \mid u_{n-1} = 0\}$ and $\{(u_{n-1}, u_{n-2}, \dots, u_0) \mid u_{n-1} = 1\}$ are both isomorphic to Q_{n-1} . In other words, B_n consists of two Q_{n-1} . In the figure, the left- and right-hand side subgraphs form two distinct Q_3 's. Furthermore, if n is even, the subgraphs induced by the vertex sets $\{(u_{n-1}, u_{n-2}, \dots, u_0) \mid u_{n-2} = 0\}$ and $\{(u_{n-1}, u_{n-2}, \dots, u_0) \mid u_{n-2} = 1\}$ are both isomorphic to B_{n-1} . In other words, B_n (n : even) consists of two B_{n-1} 's. In the figure, the upper- and lower-side subgraphs form two distinct B_3 's.

Definition 4 For a pair of vertices u and v , let $P(u, v)$ be the subset of adjacent vertices of u given by $P(u, v) = \{w \mid d(u, w) = 1, d(w, v) = d(u, v) - 1\}$. $P(u, v)$ is called the preferred adjacent vertex set of u to v . \square

In other words, $P(u, v)$ represents the adjacent vertex set that contains the adjacent vertices of u each of which is on a shortest path from u to v .

3 Comparison of Cube-Based Topologies for Interconnection Networks

In this section, we first present the most popular topology, the hypercube, and its variants. Then, we compare their properties regarding the diameter and the symmetry in a table.

The hypercube [5] was once widely adopted as a topology of many interconnection networks. Q_n has a degree n and a diameter n and connects 2^n vertices. Also, it has a good property of vertex and edge symmetry. There are many previous works regarding the hypercube. Bossard and Kaneko proposed an algorithm that generates k vertex-disjoint paths between one vertex and k ($k \leq n$) distinct vertices of lengths at most $n + 1$ in $O(k^3 + kn)$ time in Q_n [9]. Duong and Kaneko proposed two fault-tolerant routing algorithms for hypercubes based on the approximate directed routable probabilities [10].

The bicube [4] was proposed by Lim et al. B_n has a degree n and a diameter $\lceil (n + 1)/2 \rceil$ ($n \geq 7$) and connects 2^n vertices. It is not edge-symmetric but is vertex-symmetric. Unfortunately, there is not previous work regarding this topology.

The Möbius cube [1] was proposed by Cull and Larson. An n -dimensional 0-Möbius cube, $0-M_n$, has a degree n and a diameter $\lceil (n + 2)/2 \rceil$ and connects 2^n vertices. On the other hand, an n -dimensional 1-Möbius cube, $1-M_n$, has a degree n and a diameter $\lceil (n + 1)/2 \rceil$ and connects 2^n vertices. It is not vertex- or edge-symmetric. There are several previous works regarding the Möbius cube. Lim et al. proved that a complete binary tree with $2^n - 1$ vertices can be embedded into an n -dimensional Möbius cube with dilation 1, congestion 1, and load 1 [11]. Kocík et al. proposed an algorithm that generates n vertex-disjoint paths between a vertex and n vertices of lengths at most $2n - 1$ in $O(n^4)$ time [12]. Kocík and Kaneko proposed an algorithm that generates n vertex-disjoint paths between an arbitrary pair of vertices of length at most $3n - 5$ in $O(n^2)$ time [13]. Figures 3 and 4 show examples of $0-M_4$ and $1-M_4$, respectively.

The crossed cube [2] was proposed by Efe. An n -dimensional crossed cube, C_n , has a degree n and a diameter $\lceil (n+1)/2 \rceil$ and connects 2^n vertices. It is not vertex- or edge-symmetric. The crossed cube has many previous works. Satoh et al. proposed an algorithm that can classify the adjacent vertices of an arbitrary vertex in C_n into those on the shortest paths, the sidetrack paths, and the backtrack paths to the destination vertex in $O(n)$ time [14]. Cheng et al. proposed an algorithm that constructs $(n - 2)$ independent spanning trees rooted at vertex 0 in C_n [15]. Figure 5 shows an example of C_4 .

The twisted cube [3] was proposed by Hilbert et al. An n -dimensional twisted cube, T_n , has a degree n and a diameter $\lceil (n + 1)/2 \rceil$ and connects 2^n vertices. It is not vertex- or edge-symmetric. There are many previous works regarding the twisted cube. Chang et al. proved that the wide diameter and the fault diameter of T_n are both $\lceil n/2 \rceil + 2$ and the twisted cube is pancyclic, that is, it has a cycle of an arbitrary length of at least 4 [16]. Huang et al. proved that T_n with faulty vertices has a fault-free Hamiltonian cycle if the number of faulty vertices is at most $n - 2$ and has a fault-free Hamiltonian path between an arbitrary pair of non-faulty vertices if the number of faulty vertices is at most $n - 3$ [17]. Figure 6 shows T_4 .

Fig. 3 Example of $0-M_4$

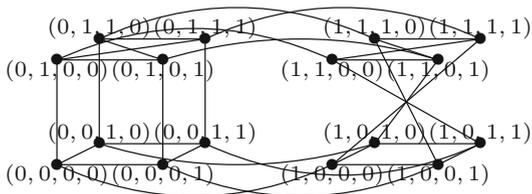


Fig. 4 Example of $1-M_4$

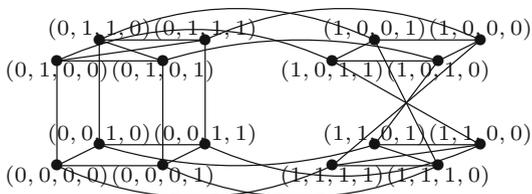


Fig. 5 Example of C_4

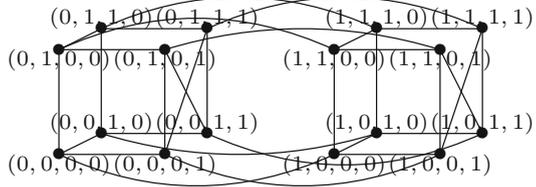


Fig. 6 Example of T_4

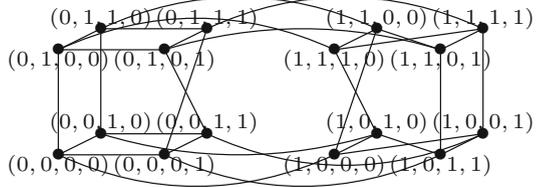


Fig. 7 Example of TC_4

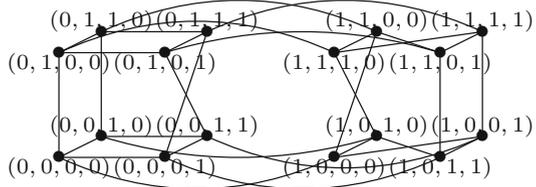
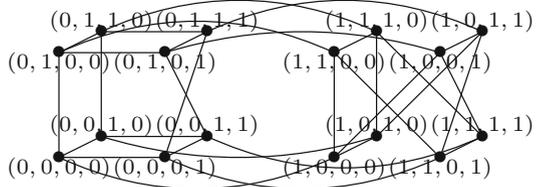


Fig. 8 Example of LT_4



The twisted crossed cube [6] was proposed by Wang et al. An n -dimensional twisted crossed cube, TC_n , has a degree n and a diameter $\lceil (n + 1)/2 \rceil$ and connects 2^n vertices. It is not vertex- or edge-symmetric. The twisted crossed cube does not have many previous works. Nagashima et al. proposed an algorithm that generates n vertex-disjoint paths of lengths at most $4n - 8$ between an arbitrary pair of vertices in TC_n in $O(n^2)$ time [18]. Figure 7 shows an example of TC_4 .

The locally twisted cube [7] was proposed by Yang et al. An n -dimensional locally twisted cube, LT_n , has a degree n and a diameter $\lceil (n + 3)/2 \rceil$ ($n \geq 5$) and connects 2^n vertices. It is not vertex- or edge-symmetric. There are many previous works with respect to the locally twisted cube. Liu et al. proved that a completely binary tree can be embedded into LT_n with dilation 2, congestion 1, load 1, and expansion 1 [19]. They also proposed three algorithms for fault-tolerant embedding of complete binary trees into faulty locally twisted cubes. Takano and Kaneko proposed a fault-tolerant routing algorithm based on two kinds of routing probabilities [20]. Figure 8 shows an example of LT_4 .

Fig. 9 Example of S_4

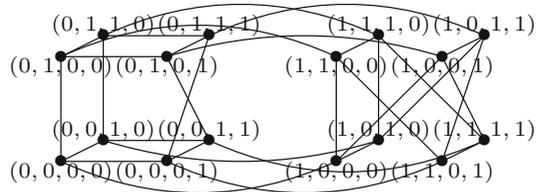


Table 1 Cube-based topologies

Topology	Diameter	Symmetry	
		Vertex	Edge
Q_n	n	✓	✓
B_n	$\lceil (n+1)/2 \rceil^a$	✓	✗
$0-M_n$	$\lceil (n+2)/2 \rceil$	✗	✗
$1-M_n$	$\lceil (n+1)/2 \rceil$	✗	✗
C_n	$\lceil (n+1)/2 \rceil$	✗	✗
T_n	$\lceil (n+1)/2 \rceil$	✗	✗
TC_n	$\lceil (n+1)/2 \rceil$	✗	✗
LT_n	$\lceil (n+3)/2 \rceil^b$	✗	✗
S_n	$\lceil n/3 \rceil + 3^c$	✗	✗

^a $n \geq 7$

^b $n \geq 5$

^c $n \geq 14$

The spined cube [8] was proposed by Zhou et al. An n -dimensional spined cube, S_n , has a degree n and a diameter $\lceil n/3 \rceil + 3$ ($n \geq 14$) and connects 2^n vertices. It is not vertex- or edge-symmetric. The spined cube does not have many previous works. Satoh et al. proposed a shortest-path routing algorithm in the spined cube [21]. Figure 9 shows an example of S_4 .

Table 1 shows the comparison of abovementioned topologies with the order 2^n and the degree n regarding the diameters and the symmetric properties. As it is shown in the table, only the bicube has the vertex-symmetric property except for the hypercube. The vertex symmetry is more important than the edge symmetry because each vertex can have a same program based on a common algorithm to process a task. Therefore, the bicube provides a promising topology for the interconnection networks of the massively parallel systems.

4 Shortest-Path Routing Algorithm

In this section, we describe our shortest-path routing algorithm in B_n ($n \geq 7$). The algorithm obtains $P(u, v)$ for the source vertex u and the destination vertex v and selects one of $P(u, v)$. By repeating this process, the algorithm arrives at the destination vertex along a shortest path. As shown in Definition 3, bicubes

have different connections between the odd and even dimensions. Therefore, the algorithm consists of two distinct approaches depending on the dimensions.

4.1 Shortest-Path Routing in an Odd-Dimensional Bicube

In B_n with an odd n , Fig. 10 shows the procedure of the shortest-path routing algorithm RO. To obtain a shortest path from a source vertex \mathbf{u} and a destination vertex \mathbf{v} , we can call the procedure with $\text{RO}(\mathbf{u}, \mathbf{v})$. RO first obtains $\mathbf{z} = \mathbf{u} \oplus \mathbf{v}$ and $k = H(\mathbf{u}, \mathbf{v})$. Next, it is divided into two cases depending on the parity of $\sum_{i=0}^{n-2} u_i$ since the bit sequence of $\mathbf{u}^{(n-1)}$ is different. Then, it is still divided into cases depending on the values of k and z_{n-1} .

In the rest of this section, we give a proof of correctness of RO and its time complexity. Note that we present some of the lemmas without their proofs.

```

procedure RO( $\mathbf{u}, \mathbf{v}$ )
 $\mathbf{z} := \mathbf{u} \oplus \mathbf{v}$ ;
 $k := \sum_{i=0}^{n-1} z_i$ ;
if  $\sum_{i=0}^{n-2} u_i$  is even then
  if  $k = n$  then return  $\{\mathbf{u}^{(i)} \mid n-2 \geq i \geq 0\}$ 
  else if  $k = n-1$  then
    if  $z_{n-1} = 0$  then return  $\{\mathbf{u}^{(i)} \mid n-1 \geq i \geq 0\}$ 
    else return  $\{\mathbf{u}^{(q)} \mid z_q = 0\}$  endif
  else if  $k > \lceil n/2 \rceil$  then
    if  $z_{n-1} = 0$  and  $k = n-2$  then return  $\{\mathbf{u}^{(n-1)}\}$ 
    else return  $\{\mathbf{u}^{(q)} \mid z_q = 0\}$  endif
  else if  $k = \lceil n/2 \rceil$  then return  $\{\mathbf{u}^{(i)} \mid n-1 \geq i \geq 0\}$ 
  else if  $z_{n-1} = 1$  and  $k \leq 2$  then return  $\{\mathbf{u}^{(n-1)}\}$ 
  else return  $\{\mathbf{u}^{(q)} \mid z_q = 1\}$  endif
else /*  $\sum_{i=0}^{n-2} u_i$  is odd */
  if  $k = n$  then return  $\{\mathbf{u}^{(n-1)}\}$ 
  else if  $k = n-1$  then
    if  $z_{n-1} = 0$  then return  $\{\mathbf{u}^{(i)} \mid n-1 \geq i \geq 0\}$ 
    else return  $\{\mathbf{u}^{(n-1)}\}$  endif
  else if  $k > \lceil n/2 \rceil$  then
    if  $z_{n-1} = 0$  and  $k = n-2$  then return  $\{\mathbf{u}^{(n-1)}\}$ 
    else if  $z_{n-1} = 0$  and  $k \leq n-3$  then return  $\{\mathbf{u}^{(q)} \mid z_q = 0\}$ 
    else return  $\{\mathbf{u}^{(n-1)}, \mathbf{u}^{(q)} \mid z_q = 0\}$  endif
  else if  $k = \lceil n/2 \rceil$  then return  $\{\mathbf{u}^{(i)} \mid n-1 \geq i \geq 0\}$ 
  else if  $z_{n-1} = 0$  then return  $\{\mathbf{u}^{(q)} \mid z_q = 1\}$ 
  else if  $z_{n-1} = 1$  and  $k = 1$  then return  $\{\mathbf{u}^{(i)} \mid n-2 \geq i \geq 0\}$ 
  else return  $\{\mathbf{u}^{(q)} \mid z_q = 1, q \neq n-1\}$  endif
endif

```

Fig. 10 Shortest-path routing algorithm RO in an odd-dimensional bicube

Lemma 1 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n with an odd n , let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is even, the distance between \mathbf{u} and \mathbf{v} is given by

$$d(\mathbf{u}, \mathbf{v}) = \begin{cases} 3 & (k = n), \\ \min\{k, 4\} & (k = n - 1, z_{n-1} = 0), \\ 2 & (k = n - 1, z_{n-1} = 1), \\ \min\{k, n - k + 1\} & (k \leq n - 2). \end{cases}$$

(Proof) By [4]. □

Lemma 2 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n with an odd n , let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, there is at most one complimentary edge in a shortest path from \mathbf{u} to \mathbf{v} .

(Proof) First, let us introduce $\mathbf{E} \in \{0, 1\}^n$ and $\mathbf{e}_i \in \{0, 1\}^n$ ($n - 1 \geq i \geq 0$), which are defined by $\mathbf{E} = (1, 1, \dots, 1)$ and $\mathbf{e}_i = (e_{n-1}, e_{n-2}, \dots, e_0)$ where $e_j = 0$ if $j \neq i$ and $e_j = 1$ if $j = i$. Next, assume that $\mathbf{u}_0 (= \mathbf{u}) \rightarrow \mathbf{u}_1 \rightarrow \dots \rightarrow \mathbf{u}_l (= \mathbf{v})$ is a shortest path from \mathbf{u} to \mathbf{v} . Then, the edge $(\mathbf{u}_{j-1}, \mathbf{u}_j)$ ($1 \leq j \leq l$) is represented by \mathbf{e}^j where $\mathbf{e}^j \in \{\mathbf{E}, \mathbf{e}_{n-1}, \mathbf{e}_{n-2}, \dots, \mathbf{e}_0\}$. That is, $\mathbf{u}_j = \mathbf{u}_{j-1} \oplus \mathbf{e}^j$. Now, assume that the shortest path contains multiple complementary edges, that is, $\{\mathbf{e}^1, \mathbf{e}^2, \dots, \mathbf{e}^l\}$ contains multiple \mathbf{E} . Then, we can assume without loss of generality that j_1 and j_2 are two smallest indices such that $\mathbf{e}^{j_1} = \mathbf{e}^{j_2} = \mathbf{E}$. Because $\mathbf{E} \oplus \mathbf{E} = \mathbf{0}$, $\bigoplus_{j=1}^{j_2} \mathbf{e}^j = \mathbf{e}^1 \oplus \mathbf{e}^2 \oplus \dots \oplus \mathbf{e}^{j_1-1} \oplus \mathbf{e}^{j_1+1} \oplus \dots \oplus \mathbf{e}^{j_2-1}$. Hence, $\mathbf{u}_{j_2} (= \mathbf{u}_0 \oplus \bigoplus_{j=1}^{j_2} \mathbf{e}^j) = \mathbf{u}_0 \oplus \mathbf{e}^1 \oplus \mathbf{e}^2 \oplus \dots \oplus \mathbf{e}^{j_1-1} \oplus \mathbf{e}^{j_1+1} \oplus \dots \oplus \mathbf{e}^{j_2-1}$. Also, because $\mathbf{e}^j \in \{\mathbf{e}_{n-1}, \mathbf{e}_{n-2}, \dots, \mathbf{e}_0\}$ for any $j \in \{1, 2, \dots, j_1 - 1, j_1 + 1, \dots, j_2 - 1\}$, an arbitrary vertex has an edge corresponding to \mathbf{e}^j . Therefore, there is a shorter path from $\mathbf{u}_0 (= \mathbf{u})$ to \mathbf{u}_{j_2} . This is a contradiction with the assumption that $\mathbf{u}_0 (= \mathbf{u}) \rightarrow \mathbf{u}_1 \rightarrow \dots \rightarrow \mathbf{u}_l (= \mathbf{v})$ is a shortest path from \mathbf{u} to \mathbf{v} . Consequently, any shortest path does not include multiple complementary edges. □

Lemma 3 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n (n : odd), let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is even and $k = n$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(i)} \mid n - 2 \geq i \geq 0\}$.

(Proof) From $k = n$, $\mathbf{v} = (\overline{u_{n-1}}, \overline{u_{n-2}}, \dots, \overline{u_0})$. From $\mathbf{u}^{(i)} (= (u'_{n-1}, u'_{n-2}, \dots, u'_0)) = (u_{n-1}, u_{n-2}, \dots, u_{i+1}, \overline{u_i}, u_{i-1}, \dots, u_0)$ ($n - 2 \geq i \geq 0$), $\sum_{i=0}^{n-2} u'_i$ is odd. Therefore, $\mathbf{u}^{(i, n-1)} = (\overline{u_{n-1}}, \overline{u_{n-2}}, \dots, \overline{u_{i+1}}, u_i, \overline{u_{i-1}}, \dots, \overline{u_0}) = \mathbf{v}^{(i)}$ holds, and $d(\mathbf{u}^{(i, n-1)}, \mathbf{v}) = 1$. Hence, $d(\mathbf{u}^{(i)}, \mathbf{v}) = 2$ holds. Also, from $\mathbf{u}^{(n-1)} = (\overline{u_{n-1}}, u_{n-2}, \dots, u_0)$ and Lemma 1, $d(\mathbf{u}^{(n-1)}, \mathbf{v}) = 4$ holds. Since $d(\mathbf{u}, \mathbf{v}) = 3$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(i)} \mid n - 2 \geq i \geq 0\}$. □

Lemma 4 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n (n : odd), let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is even, $k = n - 1$, and $z_{n-1} = 0$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(i)} \mid n - 1 \geq i \geq 0\}$.

(Proof) From $k = n - 1$ and $z_{n-1} = 0$, $\mathbf{u} = (v_{n-1}, \overline{v_{n-2}}, \dots, \overline{v_0})$. Since $\sum_{i=0}^{n-2} u_i$ is even, $\mathbf{u}^{(n-1)} = \overline{\mathbf{v}}$. From Lemma 1, $d(\mathbf{u}^{(n-1)}, \mathbf{v}) = 3$ holds. From $\mathbf{u}^{(i)} (= (u'_{n-1}, u'_{n-2}, \dots, u'_0)) = (v_{n-1}, \overline{v_{n-2}}, \dots, \overline{v_{i+1}}, v_i, \overline{v_{i-1}}, \dots, \overline{v_0})$ ($n - 2 \geq i \geq 0$), $\sum_{i=0}^{n-2} u'_i$ is odd. Hence, $\mathbf{u}^{(i,n-1)} = (\overline{v_{n-1}}, v_{n-2}, \dots, v_{i+1}, \overline{v_i}, v_{i-1}, \dots, v_0)$. From $\mathbf{u}^{(i,n-1,i)} = (\overline{v_{n-1}}, v_{n-2}, \dots, v_{i+1}, v_i, v_{i-1}, \dots, v_0) = \mathbf{v}^{(n-1)}$, $d(\mathbf{u}^{(i,n-1,i)}, \mathbf{v}) = 1$ and then $d(\mathbf{u}^{(i)}, \mathbf{v}) = 3$ holds. Therefore, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(i)} \mid n - 1 \geq i \geq 0\}$. \square

Lemma 5 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n (n : odd), let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is even, $k = n - 1$, and $z_{n-1} = 1$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(q)} \mid z_q = 0\}$.

(Proof) From $k = n - 1$ and $z_{n-1} = 1$, $\mathbf{u} = (\overline{v_{n-1}}, v_{n-2}, \dots, v_{q+1}, v_q, \overline{v_{q-1}}, \dots, \overline{v_0})$ ($n - 2 \geq q \geq 0$) holds. Here, $\sum_{i=0}^{n-2} u_i$ is odd since $\mathbf{u}^{(q)} = (\overline{v_{n-1}}, v_{n-2}, \dots, \overline{v_0})$. Hence, $d(\mathbf{u}^{(q)}, \mathbf{v}) = 1$ holds. Also, since any adjacent vertex of \mathbf{u} except for $\mathbf{u}^{(q)}$ is not adjacent to \mathbf{v} , $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(q)} \mid z_q = 0\}$ holds. \square

Lemma 6 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n (n : odd), let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is even, $k > \lceil n/2 \rceil$, $z_{n-1} = 0$, and $k = n - 2$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(n-1)}\}$.

(Proof) From $\sum_{i=0}^{n-2} u_i$ is even, $z_{n-1} = 0$, and $k = n - 2$, $\mathbf{u} = (v_{n-1}, \overline{v_{n-2}}, \dots, \overline{v_{q+1}}, v_q, \overline{v_{q-1}}, \dots, \overline{v_0})$ ($n - 2 \geq q \geq 0$). Hence, $\mathbf{u}^{(n-1)} = (\overline{v_{n-1}}, v_{n-2}, \dots, v_{q+1}, v_q, \overline{v_{q-1}}, \dots, \overline{v_0})$ holds. Here, from Lemma 1, $d(\mathbf{u}^{(n-1)}, \mathbf{v}) = 2$ holds. Also, from $k = n - 2$ and Lemma 1, $d(\mathbf{u}, \mathbf{v}) = \min\{k, n - k + 1\} = 3$ holds.

On the other hand, from $\mathbf{u}^{(q)} = (v_{n-1}, \overline{v_{n-2}}, \dots, \overline{v_0})$, $\mathbf{u}^{(q,r)} = (v_{n-1}, \overline{v_{n-2}}, \dots, \overline{v_{r+1}}, v_r, \overline{v_{r-1}}, \dots, \overline{v_0})$ ($n - 2 \geq r \geq 0$) is not adjacent to \mathbf{v} . Furthermore, $\mathbf{u}^{(q,n-1)} = (\overline{v_{n-1}}, v_{n-2}, \dots, v_0)$ is not adjacent to \mathbf{v} .

Also, from $\mathbf{u}^{(p)} = (v_{n-1}, \overline{v_{n-2}}, \dots, \overline{v_{p+1}}, v_p, \overline{v_{p-1}}, \dots, \overline{v_{q+1}}, v_q, \overline{v_{q-1}}, \dots, \overline{v_0})$ ($n - 2 \geq p \neq q \geq 0$), $\mathbf{u}^{(p,q)} = (\overline{v_{n-1}}, v_{n-2}, \dots, v_{q+1}, \overline{v_q}, \overline{v_{q-1}}, \dots, \overline{v_{p+1}}, v_p, \overline{v_{p-1}}, \dots, \overline{v_0})$, and $\mathbf{u}^{(p,r)} = (v_{n-1}, \overline{v_{n-1}}, \dots, \overline{v_{p-1}}, v_p, \overline{v_{p-1}}, \dots, \overline{v_{q+1}}, v_q, \overline{v_{q-1}}, \dots, \overline{v_{r-1}}, v_r, \overline{v_{r-1}}, \dots, \overline{v_0})$ are not adjacent to \mathbf{v} . \square

Lemma 7 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n (n : odd), let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is even, $k > \lceil n/2 \rceil$, and either $z_{n-1} = 1$ or $k \neq n - 2$ holds, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(q)} \mid z_q = 0\}$.

Lemma 8 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n (n : odd), let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is even, $k < \lceil n/2 \rceil$, $z_{n-1} = 1$, and $k \leq 2$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(n-1)}\}$.

(Proof) If $k = 1$, $\mathbf{u} = (\overline{v_{n-1}}, v_{n-2}, \dots, v_0)$ holds. From $\mathbf{u}^{(n-1)} = \mathbf{v}$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(n-1)}\}$ holds. If $k = 2$, from $\mathbf{u} = (\overline{v_{n-1}}, v_{n-2}, \dots, v_{q+1}, \overline{v_q}, v_{q-1}, \dots, v_0)$, $\mathbf{u}^{(n-1)} = (v_{n-1}, v_{n-2}, \dots, v_{q+1}, \overline{v_q}, v_{q-1}, \dots, v_0)$. Hence, $d(\mathbf{u}^{(n-1)}, \mathbf{v}) = 1$. On the other hand, any adjacent vertex of \mathbf{u} except for $\mathbf{u}^{(n-1)}$ is not adjacent to \mathbf{v} . From $d(\mathbf{u}, \mathbf{v}) = 2$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(n-1)}\}$. \square

Lemma 9 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n (n : odd), let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is even, $k < \lceil n/2 \rceil$, and either $z_{n-1} = 0$ or $k \geq 3$ holds, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(q)} \mid z_q = 1\}$.

(Proof) If $z_{n-1} = 0$, \mathbf{u} and \mathbf{v} belong to a subgraph that is isomorphic to Q_{n-1} . From $k < \lceil n/2 \rceil$, it is possible to find a shortest path as a hypercube. Hence, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(q)} \mid z_q = 1\}$.

If $z_{n-1} = 1$ and $k \geq 3$, $\mathbf{u}^{(n-1)}$ and \mathbf{v} belong to a subgraph that is isomorphic to Q_{n-1} . Hence, from the similar discussion above, $\mathbf{u}^{(n-1)} \in P(\mathbf{u}, \mathbf{v})$ holds. Moreover, from $\lceil n/2 \rceil > k \geq 3$, $d(\mathbf{u}, \mathbf{v}) = \min\{k, n - k + 1\} = k$ holds from Lemma 1. Hence, the shortest path does not include any complementary edge. Therefore, $H(\mathbf{u}^{(q)}, \mathbf{v}) = k - 1$ ($z_q = 1, q \neq n - 1$) holds. By selecting the $(n - 1)$ -th adjacent vertex such that the path does not include any complementary edge, it is possible to construct the path of length $k - 1$ from $\mathbf{u}^{(q)}$ ($z_q = 1, q \neq n - 1$) to \mathbf{v} . Therefore, $\{\mathbf{u}^{(q)} \mid z_q = 1, q \neq n - 1\} \subset P(\mathbf{u}, \mathbf{v})$. On the other hand, for any vertex $\mathbf{u}^{(r)} \in \{\mathbf{u}^{(r)} \mid z_q = 0, r \neq q\}$, $H(\mathbf{u}^{(r)}, \mathbf{v}) = k + 1$. Because the path does not include any complementary edge, $d(\mathbf{u}^{(r)}, \mathbf{v}) \geq k + 1$ holds.

From the discussion above, if $\sum_{i=0}^{n-2} u_i$ is even, $k < \lceil n/2 \rceil$, and either $z_{n-1} = 0$ or $k \geq 3$ holds, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(q)} \mid z_q = 1\}$. □

Lemma 10 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n (n : odd), let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is even, $k = \lceil n/2 \rceil$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(i)} \mid n - 1 \geq i \geq 0\}$.

Lemma 11 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n (n : odd), let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is odd and $k = n$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(n-1)}\}$.

(Proof) From $k = n$, $\mathbf{u} = (\overline{v_{n-1}}, \overline{v_{n-2}}, \dots, \overline{v_0})$ holds. Then, because $\sum_{i=0}^{n-2} u_i$ is odd, $\mathbf{u}^{(n-1)} = \mathbf{v}$. Therefore, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(n-1)}\}$. □

Lemma 12 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n (n : odd), let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is odd, $k = n - 1$, and $z_{n-1} = 0$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(i)} \mid n - 1 \geq i \geq 0\}$.

(Proof) From $k = n - 1$ and $z_{n-1} = 0$, $\mathbf{u} = (v_{n-1}, \overline{v_{n-2}}, \dots, \overline{v_0})$ holds. Because $\sum_{i=0}^{n-2} u_i$ is odd, $\mathbf{u}^{(n-1)} = (\overline{v_{n-1}}, v_{n-2}, \dots, v_0)$, and $\mathbf{u}^{(n-1, q)} = (\overline{v_{n-1}}, v_{n-2}, \dots, v_{q+1}, \overline{v_q}, v_{q-1}, \dots, v_0)$ ($n - 2 \geq q \geq 0$) hold. Because $\mathbf{u}^{(n-1, q, n-1)} = (v_{n-1}, v_{n-2}, \dots, v_{q+1}, \overline{v_q}, v_{q-1}, \dots, v_0)$, $d(\mathbf{u}^{(n-1, q, n-1)}, \mathbf{v}) = 1$ and $d(\mathbf{u}^{(n-1)}, \mathbf{v}) = 3$ hold. In addition, $\{\mathbf{u}^{(i)} \mid n - 2 \geq i \geq 0\} = \{(v_{n-1}, \overline{v_{n-2}}, \dots, \overline{v_{i+1}}, v_i, \overline{v_{i-1}}, \dots, \overline{v_0}) \mid n - 2 \geq i \geq 0\}$ holds. Hence, $d(\mathbf{u}^{(i)}, \mathbf{v}) = 3$ ($n - 2 \geq i \geq 0$) from Lemma 1. From above discussion, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(i)} \mid n - 1 \geq i \geq 0\}$. □

Lemma 13 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n (n : odd), let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is odd, $k = n - 1$, and $z_{n-1} = 1$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(n-1)}\}$.

(Proof) From $k = n - 1$ and $z_{n-1} = 1$, $\mathbf{u} = (\overline{v_{n-1}}, \overline{v_{n-2}}, \dots, \overline{v_{q+1}}, \overline{v_q}, \overline{v_{q-1}}, \dots, \overline{v_0})$ ($n - 2 \geq q \geq 0$) holds. Then, $\mathbf{u}^{(n-1)} = (v_{n-1}, v_{n-2}, \dots, v_{q+1}, \overline{v_q}, \overline{v_{q-1}}, \dots, \overline{v_0})$

$(n - 2 \geq q \geq 0)$. Hence, $\mathbf{u}^{(n-1)}$ and \mathbf{v} are adjacent. On the other hand, $\mathbf{u}^{(i)}$ $(n - 2 \geq i \geq 0)$ are not adjacent to \mathbf{v} . Hence, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(n-1)}\}$. \square

Lemma 14 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n (n : odd), let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is odd, $k > \lceil n/2 \rceil$, $z_{n-1} = 0$, and $k = n - 2$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(n-1)}\}$.

(Proof) Because $\sum_{i=0}^{n-2} u_i$ is odd, $z_{n-1} = 0$, and $k = n - 2$, $\mathbf{u} = (\overline{v_{n-1}}, \overline{v_{n-2}}, \dots, \overline{v_{q+1}}, v_q, \overline{v_{q-1}}, \dots, \overline{v_0})$ $(n - 2 \geq q \geq 0)$. Hence, $\mathbf{u}^{(n-1)} = (\overline{v_{n-1}}, v_{n-2}, \dots, v_{q+1}, \overline{v_q}, \overline{v_{q-1}}, \dots, \overline{v_0})$ holds. Then, $\mathbf{u}^{(n-1, q)} = (\overline{v_{n-1}}, v_{n-2}, \dots, v_0)$ holds, and $\mathbf{u}^{(n-1, q)}$ and \mathbf{v} are adjacent. Therefore, $d(\mathbf{u}^{(n-1)}, \mathbf{v}) = 2$. On the other hand, $\mathbf{u}^{(q)} = (v_{n-1}, \overline{v_{n-2}}, \dots, \overline{v_0})$. Hence, $d(\mathbf{u}^{(q)}, \mathbf{v}) = 4$ holds from Lemma 1. In addition, for any adjacent vertex of \mathbf{u} other than $\mathbf{u}^{(n-1)}$ and $\mathbf{u}^{(q)}$, the distance between the vertex and \mathbf{v} is 4 from Lemma 1. Consequently, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(n-1)}\}$. \square

Lemma 15 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n (n : odd), let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is odd, $k > \lceil n/2 \rceil$, $z_{n-1} = 0$, and $k \leq n - 3$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(q)} \mid z_q = 0\}$.

(Proof) For $\mathbf{u}^{(q)} = (u'_{n-1}, u'_{n-2}, \dots, u'_0)$ ($z_q = 0, q \neq n - 1$), $H(\mathbf{u}^{(n-1)}, \mathbf{v}) = n - k$ holds because $\sum_{i=0}^{n-2} u'_i$ is odd. Also, $H(\mathbf{u}^{(n-1, q)}, \mathbf{v}) = n - k - 1$ ($z_q = 1, q \neq n - 1$) holds. From $k > \lceil n/2 \rceil$, $n - k - 1 < \lceil n/2 \rceil - 1$. Hence, from Lemma 1, $d(\mathbf{u}^{(n-1, q)}, \mathbf{v}) = \min\{n - k - 1, k + 2\} = n - k - 1$ holds. Therefore, $d(\mathbf{u}^{(n-1)}, \mathbf{v}) = n - k$.

For $\mathbf{u}^{(q)} = (u'_{n-1}, u'_{n-2}, \dots, u'_0)$ ($z_q = 0, q \neq n - 1$), $\sum_{i=0}^{n-2} u'_i$ is even. $H(\mathbf{u}^{(q)}, \mathbf{v}) = k + 1$ and $n - 3 \geq k \geq \lceil n/2 \rceil$ hold. Hence, $d(\mathbf{u}^{(q)}, \mathbf{v}) = \min\{k + 1, n - k\} = n - k$ from Lemma 1.

For $\mathbf{u}^{(r)} = (u'_{n-1}, u'_{n-2}, \dots, u'_0)$ ($z_r = 1$), $\sum_{i=0}^{n-2} u'_i$ is even. $H(\mathbf{u}^{(r)}, \mathbf{v}) = k - 1$ and $n - 3 \geq k > \lceil n/2 \rceil$. Hence, $d(\mathbf{u}^{(r)}, \mathbf{v}) = \min\{k - 1, n - k + 2\} = n - k + 2$ from Lemma 1.

From the above discussion, if $\sum_{i=0}^{n-2} u_i$ is odd, $k > \lceil n/2 \rceil$, $z_{n-1} = 0$, and $k \leq n - 3$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(q)} \mid z_q = 0\}$. \square

Lemma 16 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n (n : odd), let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is odd, $k > \lceil n/2 \rceil$, and $z_{n-1} = 1$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(n-1)}, \mathbf{u}^{(q)} \mid z_q = 0\}$.

(Proof) For $\mathbf{u}^{(q)} = (u'_{n-1}, u'_{n-2}, \dots, u'_0)$ ($z_q = 0, q \neq n - 1$), $H(\mathbf{u}^{(n-1)}, \mathbf{v}) = n - k$ holds because $\sum_{i=0}^{n-2} u'_i$ is odd. From $k > \lceil n/2 \rceil$, $n - k < \lfloor n/2 \rfloor$ holds. Also, $\mathbf{u}^{(n-1)}$ and \mathbf{v} belong to a subgraph that is isomorphic to Q_{n-1} . Hence, it is possible to find a shortest path as a hypercube, and $d(\mathbf{u}^{(n-1)}, \mathbf{v}) = n - k$.

For $\mathbf{u}^{(q)} = (u'_{n-1}, u'_{n-2}, \dots, u'_0)$ ($z_q = 0$), $\sum_{i=0}^{n-2} u'_i$ is even. Also, $H(\mathbf{u}^{(q)}, \mathbf{v}) = k + 1$ holds. Then, from $n - 2 \geq k > \lceil n/2 \rceil$, if $k = n - 2$, $d(\mathbf{u}^q, \mathbf{v}) = 2 = n - k$ holds from Lemma 1.

For $\mathbf{u}^{(r)} = (u'_{n-1}, u'_{n-2}, \dots, u'_0)$ ($z_r = 1, r \neq n - 1$), $\sum_{i=0}^{n-2} u'_i$ is even. Also, $H(\mathbf{u}^{(r)}, \mathbf{v}) = \min\{k - 1, n - k + 2\} = n - k + 2 > n - k$ holds.

From the above discussion, if $\sum_{i=0}^{n-2} u_i$ is odd, $k > \lceil n/2 \rceil$, and $z_{n-1} = 1$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(n-1)}, \mathbf{u}^{(q)} \mid z_q = 0\}$. □

Lemma 17 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n (n : odd), let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is odd, $k < \lceil n/2 \rceil$, and $z_{n-1} = 0$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(q)} \mid z_q = 1\}$.
 (Proof) From $z_{n-1} = 0$, \mathbf{u} and \mathbf{v} belong to a same subgraph that is isomorphic to Q_{n-1} . From $k < \lceil n/2 \rceil$, it is possible to find a shortest path as a hypercube. Therefore, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(q)} \mid z_q = 1\}$. □

Lemma 18 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n (n : odd), let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is odd, $k < \lceil n/2 \rceil$, $z_{n-1} = 1$, and $k = 1$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(i)} \mid n - 2 \geq i \geq 0\}$.

(Proof) From $z_{n-1} = 1$ and $k = 1$, $\mathbf{u} = (\overline{v_{n-1}}, v_{n-2}, \dots, v_0)$. Then, $\mathbf{u}^{(i)} = (\overline{v_{n-1}}, v_{n-2}, \dots, v_{i+1}, \overline{v_i}, v_{i-1}, \dots, v_0)$ ($n-2 \geq i \geq 0$). Because $\mathbf{u}^{(i, n-1)} = (v_{n-1}, v_{n-2}, \dots, v_{i+1}, \overline{v_i}, v_{i-1}, \dots, v_0)$, $\mathbf{u}^{(i, n-1)}$ and \mathbf{v} are adjacent. Hence, $d(\mathbf{u}^{(i)}, \mathbf{v}) = 2$. In addition, because $\sum_{i=0}^{n-2} u_i$ is odd, $\mathbf{u}^{(n-1)} = (v_{n-1}, \overline{v_{n-2}}, \dots, \overline{v_0})$ holds. Then, $\mathbf{u}^{(n-1, i)} = (v_{n-1}, \overline{v_{n-2}}, \dots, v_{i+1}, v_i, \overline{v_{i-1}}, \dots, \overline{v_0})$ ($n - 2 \geq i \geq 0$) holds. From $H(\mathbf{u}^{(n-1, i)}, \mathbf{v}) = n - 2$ and Lemma 1, $d(\mathbf{u}^{(n-1)}, \mathbf{v}) = 4$ holds. From the above discussion, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(i)} \mid n - 2 \geq i \geq 0\}$. □

Lemma 19 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n (n : odd), let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is odd, $k < \lceil n/2 \rceil$, $z_{n-1} = 1$, and $k \geq 2$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(q)} \mid z_q = 1, q \neq n - 1\}$.

Lemma 20 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n (n : odd), let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is odd and $k = \lceil n/2 \rceil$, $P(\mathbf{u}, \mathbf{v}) = \{\mathbf{u}^{(i)} \mid n - 1 \geq i \geq 0\}$.

Theorem 1 For two vertices \mathbf{u} and \mathbf{v} in B_n (n : odd), RO gives the preferred adjacent vertex set of \mathbf{u} to \mathbf{v} , $P(\mathbf{u}, \mathbf{v})$, in time complexity $O(n)$.

(Proof) From Lemmas 3 to 20, it is proved that RO gives $P(\mathbf{u}, \mathbf{v})$. z and k can be computed in $O(n)$ time. It takes $O(1)$ time for classification. Also, for returning $P(\mathbf{u}, \mathbf{v})$, it is enough to return the indices i of $\mathbf{u}^{(i)}$ by using an array. Hence, it takes $O(n)$ time. Therefore, RO takes $O(n)$ time in total to give $P(\mathbf{u}, \mathbf{v})$. □

Theorem 2 For two vertices $\mathbf{u} = (u_{n-1}, u_{n-2}, \dots, u_0)$ and $\mathbf{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ in B_n (n : odd), let $\mathbf{z} = (z_{n-1}, z_{n-2}, \dots, z_0) = \mathbf{u} \oplus \mathbf{v}$ and $k = \sum_{i=0}^{n-1} z_i$. Then, if $\sum_{i=0}^{n-2} u_i$ is odd, the distance between \mathbf{u} and \mathbf{v} , $d(\mathbf{u}, \mathbf{v})$, is given by

$$d(\mathbf{u}, \mathbf{v}) = \begin{cases} 1 & (k = n) \\ \min\{k, 4\} & (k = n - 1, z_{n-1} = 0) \\ 2 & (k = n - 1, z_{n-1} = 1) \\ 3 & (k = 1, z_{n-1} = 1) \\ \min\{k, n - k + 1\} & (\text{otherwise}). \end{cases}$$

(Proof) It is trivial from Lemmas 11 to 20. □

For example, for $\mathbf{u} = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ and $\mathbf{v} = (0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1)$ in B_{11} , $\text{RO}(\mathbf{u}, \mathbf{v})$ returns $\{\mathbf{u}^{(10)} = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), \mathbf{u}^{(9)} = (0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), \mathbf{u}^{(4)} = (0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0), \mathbf{u}^{(1)} = (0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0)\}$ from Lemma 7. Among the adjacent vertices, let us assume that $\mathbf{u}^{(9)}$ is selected. Next, $\text{RO}(\mathbf{u}^{(9)}, \mathbf{v})$ returns $\{\mathbf{u}^{(9,10)} = (1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1), \mathbf{u}^{(9,4)} = (0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0), \mathbf{u}^{(9,1)} = (0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0)\}$ from Lemma 15. Among the adjacent vertices, let us assume that $\mathbf{u}^{(9,10)}$ is selected. Then, $\text{RO}(\mathbf{u}^{(9,10)}, \mathbf{v})$ returns $\{\mathbf{u}^{(9,10,4)} = (1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1), \mathbf{u}^{(9,10,1)} = (1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1)\}$ from Lemma 19. Among the adjacent vertices, let us assume that $\mathbf{u}^{(9,10,1)}$ is selected. Again, $\text{RO}(\mathbf{u}^{(9,10,1)}, \mathbf{v})$ returns a singleton set $\{\mathbf{u}^{(9,10,1,10)} = (0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1)\}$. Finally, $\text{RO}(\mathbf{u}^{(9,10,1,10)}, \mathbf{v})$ returns a singleton set $\{\mathbf{u}^{(9,10,1,10,4)} = (0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1)(= \mathbf{v})\}$, and the routing terminates.

4.2 Shortest-Path Routing in an Even-Dimensional Bicube

In B_n with an even n , Fig. 11 shows the procedure of the shortest-path routing algorithm RE. To obtain a shortest path from a source vertex \mathbf{u} and a destination vertex \mathbf{v} , we can call the procedure with $\text{RE}(\mathbf{u}, \mathbf{v})$. RE first obtains $\mathbf{z} = \mathbf{u} \oplus \mathbf{v}$. Next, if $z_{n-2} = 0$, \mathbf{u} and \mathbf{v} belong to a same subgraph that is isomorphic to B_{n-1} . Hence, RE can obtain the preferred adjacent vertex set P by invoking RO by $\text{RO}((u_{n-1}, u_{n-3}, \dots, u_0), (v_{n-1}, v_{n-3}, \dots, v_0))$. Then, $\text{RE}(\mathbf{u}, \mathbf{v}) = P$. On the other hand, if $z_{n-2} = 1$, $\text{RE}(\mathbf{u}, \mathbf{v}) = P \cup \{\mathbf{u}^{(n-2)}\}$.

```

procedure RE(u, v)
z := u ⊕ v;
P0 := RO((u_{n-1}, u_{n-3}, ..., u_0), (v_{n-1}, v_{n-3}, ..., v_0));
P := {(x_{n-1}, u_{n-2}, x_{n-3}, ..., x_0) | (x_{n-1}, x_{n-3}, ..., x_0) ∈ P0};
if z_{n-2} = 0 then return P
else return P ∪ {u^{(n-2)}}
endif
    
```

Fig. 11 Shortest-path routing algorithm RE in an even-dimensional bicube

For example, for $\mathbf{u} = (0, 0, 0, 0, 1, 1, 0, 0)$ and $\mathbf{v} = (1, 0, 1, 1, 0, 1, 0, 0)$ in B_8 , P_0 returned by $\text{RO}((u_7, u_5, \dots, u_0) = (0, 0, 0, 1, 1, 0, 0), (v_7, v_5, \dots, v_0) = (1, 1, 1, 0, 1, 0, 0))$ induces $P = \{\mathbf{u}^{(7)}, \mathbf{u}^{(5)}, \mathbf{u}^{(4)}, \mathbf{u}^{(3)}, \mathbf{u}^{(2)}, \mathbf{u}^{(1)}, \mathbf{u}^{(0)}\}$. Because $z_6 = 0$, $\text{RE}(\mathbf{u}, \mathbf{v})$ returns P . For each vertex in P , there is a shortest path that includes it such as $\mathbf{u} \rightarrow \mathbf{u}^{(7)} = (1, 0, 0, 0, 1, 1, 0, 0) \rightarrow \mathbf{u}^{(7,5)} = (1, 0, 1, 0, 1, 1, 0, 0) \rightarrow \mathbf{u}^{(7,5,4)} = (1, 0, 1, 1, 1, 1, 0, 0) \rightarrow \mathbf{u}^{(7,5,4,3)} = (1, 0, 1, 1, 0, 1, 0, 0)(= \mathbf{v})$, $\mathbf{u} \rightarrow \mathbf{u}^{(5)} = (0, 0, 1, 0, 1, 1, 0, 0) \rightarrow \mathbf{u}^{(5,4)} = (0, 0, 1, 1, 1, 1, 0, 0) \rightarrow \mathbf{u}^{(5,4,7)} = (1, 0, 1, 1, 1, 1, 0, 0) \rightarrow \mathbf{u}^{(5,4,7,3)} = (1, 0, 1, 1, 0, 1, 0, 0)(= \mathbf{v})$, $\mathbf{u} \rightarrow \mathbf{u}^{(0)} = (0, 0, 0, 0, 1, 1, 0, 1) \rightarrow \mathbf{u}^{(0,7)} = (1, 0, 1, 1, 0, 0, 1, 0) \rightarrow \mathbf{u}^{(0,7,2)} = (1, 0, 1, 1, 0, 1, 1, 0) \rightarrow \mathbf{u}^{(0,7,2,1)} = (1, 0, 1, 1, 0, 1, 0, 0)(= \mathbf{v})$, and so on.

Theorem 3 can be proved in a similar way to Theorem 1.

Theorem 3 For two vertices \mathbf{u} and \mathbf{v} in B_n (n : even), RE gives the preferred adjacent vertex set of \mathbf{u} to \mathbf{v} , $P(\mathbf{u}, \mathbf{v})$, in time complexity $O(n)$.

5 Conclusion and Future Work

In this paper, we have proposed a shortest-path routing algorithm in bicubes. Also, we have theoretically proved the correctness of the algorithm. Moreover, at each vertex in an n -dimensional bicube, we have proved that the time complexity of algorithm to obtain the preferred adjacent vertices is $O(n)$.

Because a massively parallel system includes quite a few processors and links, the existence of the faulty elements is inevitable, and it is impossible to operate the system without allowing the faulty elements. Therefore, we should design algorithms for the massively parallel system so that they can tolerate faulty elements. As a future work, it is interesting for us to apply our algorithm to fault-tolerant routing in bicubes.

Acknowledgments The authors would like to express special thanks to the reviewers for their insightful comments and suggestions. This study was partly supported by a Grant-in-Aid for Scientific Research (C) of the Japan Society for the Promotion of Science under Grant No. 20K11729.

References

1. P. Cull, S.M. Larson, The Möbius cubes. *IEEE Trans. Comput.* **44**(5), 647–659 (1995)
2. K. Efe, The crossed cube architecture for parallel computing. *IEEE Trans. Parallel Distrib. Syst.* **3**(5), 513–524 (1992)
3. P.A.J. Hilbers, M.R. Koopman, J.L.A. van de Snepscheut, The twisted cube, in *Volume 1: Parallel Architectures on PARLE: Parallel Architectures and Languages Europe* (Springer, London, 1987), pp. 152–159. <http://dl.acm.org/citation.cfm?id=25489.25499>

4. H.S. Lim, J.H. Park, H.C. Kim, The bicube: an interconnection of two hypercubes. *Int. J. Comput. Math.* **92**(1), 29–40 (2015)
5. C.L. Seitz, The cosmic cube. *Commun. ACM* **28**(1), 22–33 (1985)
6. X. Wang, J. Liang, D. Qi, W. Lin, The twisted crossed cube. *Concurrency Comput. Pract. Exp.* **28**, 1507–1526 (2016)
7. X. Yang, D.J. Evans, G.M. Megson, The locally twisted cubes. *Int. J. Comput. Math.* **82**(4), 401–413 (2005)
8. W.J. Zhou, J.X. Fan, X.H. Jia, S.K. Zhang, The spined cube: a new hypercube variant with smaller diameter. *Inf. Process. Lett.* **111**(12), 561–567 (2011)
9. A. Bossard, K. Kaneko, Time optimal node-to-set disjoint paths routing in hypercubes. *J. Inf. Sci. Eng.* **30**(4), 1087–1093 (2014)
10. D.T. Duong, K. Kaneko, Fault-tolerant routing based on approximate directed routable probabilities for hypercubes. *Futur. Gener. Comput. Syst.* **37**, 88–96 (2014). <https://doi.org/10.1016/j.future.2013.12.003>. <http://www.sciencedirect.com/science/article/pii/S0167739X13002677>
11. Z. Liu, J. Fan, X. Jia, Complete binary trees embeddings in Möbius cubes. *J. Comput. Syst. Sci.* **82**(2), 260–281 (2016). <https://doi.org/10.1016/j.jcss.2015.09.004>. <http://www.sciencedirect.com/science/article/pii/S0022000015001026>
12. D. Kocík, Y. Hirai, K. Kaneko, Node-to-set disjoint paths problem in a Möbius cube. *IEICE Trans. Inf. Syst.* **E99-D**(3), 708–713 (2016)
13. D. Kocík, K. Kaneko, Node-to-node disjoint paths problem in a Möbius cube. *IEICE Trans. Inf. Syst.* **E100-D**(8), 1837–1843 (2017)
14. K. Satoh, K. Mouri, K. Kaneko, A fully adaptive minimal routing algorithm in a crossed cube, in *Proceedings of 2018 International Conference on Parallel and Distributed Processing Techniques and Applications* (2018), pp. 183–189
15. B. Cheng, J. Fan, Q. Lyu, J. Zhou, Z. Liu, Constructing independent spanning trees with height n on the n -dimensional crossed cube. *Futur. Gener. Comput. Syst.* **87**, 404–415 (2018). <https://doi.org/10.1016/j.future.2018.02.010>. <http://www.sciencedirect.com/science/article/pii/S0167739X18302723>
16. C.P. Chang, J.N. Wang, L.H. Hsu, Topological properties of twisted cube. *Inf. Sci.* **113**(1–2), 147–167 (1999)
17. W.T. Huang, J.J. Tan, C.N. Hung, L.H. Hsu, Fault-tolerant hamiltonicity of twisted cubes. *J. Parallel Distrib. Comput.* **62**(4), 591–604 (2002). <https://doi.org/10.1006/jpdc.2001.1813>. <http://www.sciencedirect.com/science/article/pii/S0743731501918131>
18. H. Nagashima, K. Mouri, K. Kaneko, Node-to-node disjoint paths in twisted crossed cubes, in *Proceedings of the 10th International Conference on Advances in Information Technology* (2018), pp. 5:1–5:8. <https://doi.org/10.1145/3291280.3291785>
19. Z. Liu, J. Fan, J. Zhou, B. Cheng, X. Jia, Fault-tolerant embedding of complete binary trees in locally twisted cubes. *J. Parallel Distrib. Comput.* **101**, 69–78 (2017). <https://doi.org/10.1016/j.jpdc.2016.11.005>. <http://www.sciencedirect.com/science/article/pii/S0743731516301496>
20. Y. Takano, K. Kaneko, Stochastic fault-tolerant routing in locally twisted cubes, in *Proceedings of the 2017 6th ICT International Student Project Conference*, pp. 1–4 (2017). <https://doi.org/10.1109/ICT-ISPC.2017.8075317>
21. K. Satoh, K. Kaneko, P.T.H. Hanh, H.T.T. Binh, Shortest-path routing in spined cubes, in *Proceedings of the 2017 6th ICT International Student Project Conference* (2017), pp. 1–4. <https://doi.org/10.1109/ICT-ISPC.2017.8075349>

An NPGA-II-Based Multi-objective Edge Server Placement Strategy for IoV



Xuan Yan, Zhanyang Xu, Mohammad R. Khosravi, Lianyong Qi, and Xiaolong Xu

1 Introduction

In Internet of Vehicles (IoV), the vehicle tasks are offloaded from the vehicles to the nearby roadside units (RSUs) for calculation. The RSUs transfer the tasks to the nearest servers and analyze the tasks on the servers [10]. The scale of these tasks has maintained increasing at a high speed on account of the expanding of IoV. Generally, cloud computing is applied to analyze the ever-increasing tasks and respond to the tasks from the RSUs. However, sending all the tasks to the remote cloud platform for execution brings tremendous bandwidth pressure and high waiting time of the tasks [15]. Compared with cloud computing, the tasks are processed near the users in edge computing, which results in the better service experience of the vehicle users. Therefore, it makes sense to adopt edge computing in IoV.

X. Yan

School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China

e-mail: xuanyan@nuist.edu.cn

Z. Xu · X. Xu (✉)

School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China

Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing, China

e-mail: zhanyang_xu@nuist.edu.cn

M. R. Khosravi

Department of Computer Engineering, Persian Gulf University, Bushehr, Iran

Department of Electrical and Electronic Engineering, Shiraz University of Technology, Shiraz, Iran

L. Qi

School of Information Science and Engineering, Qufu Normal University, Jining, China

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_39

541

With edge computing, the tasks are transmitted to the nearest edge server (ES) for analysis to reduce the time of data transmission [11]. The locations and the service performance of the ESs are closely related in IoV [12]. Since an ES often responds to the tasks from multiple RSUs, the improper locations may cause the ESs to be overloaded or underloaded. The overload and underload worsen the waiting time of the tasks and the coverage rate of the ESs, respectively. As a result, for improving workload balancing, the locations should be determined according to the distribution of the RSUs [13]. In addition, the computing capacity and the scale of the ESs are usually limited by the budget [2]. Therefore, some tasks are supposed to be processed on the cloud. To improve the service experience of the vehicle users, increasing the coverage rate of the placement scheme is also vital in IoV.

To raise the coverage rate, the proportion of tasks processed on the ESs is supposed to be high, which causes the deteriorating of the workload variance of the ESs [4]. Since the ESs are resource-limited, the additional tasks should keep waiting until the previous tasks are finished. Processing more tasks on the ESs would increase the waiting time of the tasks [7]. Similarly, improving the waiting time of the tasks by uploading some tasks to the cloud must be at the expense of coverage rate decrease. Ameliorating the workload variance by transferring the task to the distant ESs increases the waiting time of the tasks. Therefore, how to balance the coverage rate, the workload variance, and the waiting time of the tasks is a crucial problem in ES placement. In this paper, an NPGA-II-based multi-objective edge server placement strategy, named NMEPS, is designed for achieving the balance among the coverage rate, the workload variance, and the waiting time of the tasks. The main contributions are shown as follows:

- Devise the coverage rate model, workload variance model, and waiting time model and define the ES placement problem as a standard multi-objective optimization problem.
- Adopt niched Pareto genetic algorithm II (NPGA-II) and roulette algorithm to generate the feasible schemes for ES placement.
- Conduct extensive experiments by using the real-world vehicle big data to assess the validity and effectiveness of the proposed method.

In the following part of this paper, Sect. 2 introduces the related work. Sections 3 and 4 show the models and the details of the strategy, respectively. The experiment evaluation is presented in Sect. 5. In Sect. 6, we summarize the article and look forward some future work.

2 Related Work

Profit from the existence of edge computing, the distance between the RSUs and the servers becomes shorter, which causes the lower waiting time and transmitting time of the tasks [3]. Recently, a lot of work has been carried out to evaluate the benefits of edge computing.

Lin et al. [8] do research into the comprehensive perspective of computing offloading and emphasize the features of edge computing. They also illustrate some scenarios which are suitable for edge computing and look forward the future directions. Ren et al. [9] investigate the collaboration between cloud computing and edge computing. He et al. [6] propose a method named EUAGame to formulate the edge user allocation problem as a potential game. An algorithm for seeking out a Nash equilibrium in this game is designed to obtain the final results.

Multi-objective algorithms are in common use when solving the placement problems. Bakar et al. [1] applied multi-objective optimization algorithm to locate thyristor controlled series capacitor on the most sensitive line in the network. There are a great deal of algorithms for solving multi-objective optimization problem. In [14], Zan et al. used NSGA-II to optimize the locations of sensors. The problems which may be different to figure out the best solution are solved by using multi-objective algorithms. The optimal solutions found by multi-objective algorithms are acceptable in many cases [5].

To the best of our knowledge, little work paid attention to maintaining the diversity of the individuals during the evolution part in the multi-objective algorithms. Our work mainly focuses on obtaining the diverse results of the ES placement with the constant number of ESs and using roulette algorithm to accelerate convergence at the end of the evolution.

3 System Model and Problem Formulation

3.1 Network Model

Figure 1 shows a framework for IoV system in edge-cloud computing. As presented in Fig. 1, M RSUs which are denoted as (V_1, V_2, \dots, V_M) acquire data from the device it loads. These RSUs could transfer data among each other wirelessly. There are k ESs placed on the locations of k different RSUs in the overall system, where k is a constant. An RSU can transfer its data to the ESs for analyzing or upload the data to the cloud directly.

The layout of RSUs could be considered as an undirected graph, i.e., $G(v, E)$. The Euclidean distance between $V_j(x_j, y_j)$ and $V_p(x_p, y_p)$ is calculated by

$$d(V_j, V_p) = \sqrt{(x_j - x_p)^2 + (y_j - y_p)^2}, \quad (1)$$

where (x_j, y_j) represents the coordinate of V_j and (x_p, y_p) represents the coordinate of V_p .

The maximum transmission distance between RSUs is set as r_{\max} . If $d(V_j, V_p) \leq r_{\max}$, data can be transferred wirelessly from V_j to V_p , and there is an edge (V_j, V_p) in $G(v, E)$. The weight of the edge is calculated by

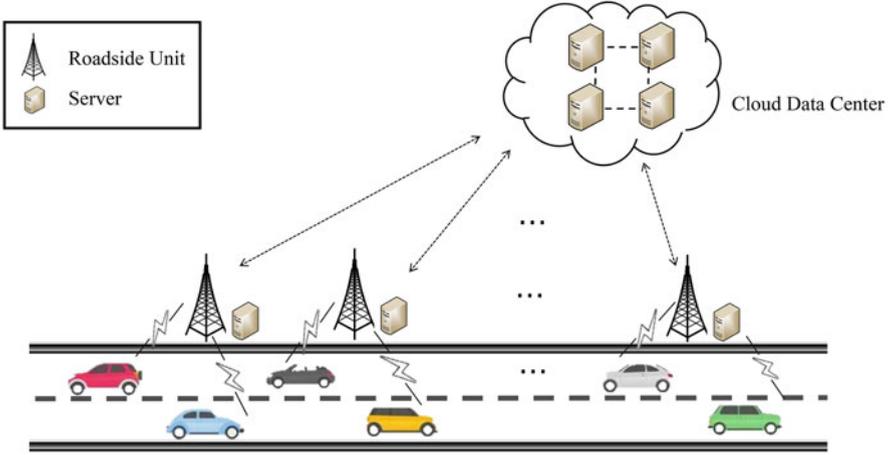


Fig. 1 A framework for IoV system in edge-cloud computing

$$WT(V_j, V_p) = \begin{cases} 0, & j = p, \\ 1, & \text{otherwise.} \end{cases} \quad (2)$$

Therefore, v is the collection of all the RSUs in $G(v, E)$ which are indexed by (V_1, V_2, \dots, V_M) , and E is made up of (V_j, V_p) , where $d(V_j, V_p) \leq r_{\max}$.

In order to alleviate the workload and bandwidth pressure of the cloud, most of the data collected by RSUs will be transmitted to the nearest ES through other RSUs. The ES analyzes the received data locally and uploads the results to the cloud.

3.2 Coverage Model

Determining the target object of data transmission entirely by observation results will not always be the proper choice. On the one hand, in $G(v, E)$, there may be a RSU denoted as V_j which is too remote or even inaccessible to any other RSU. In this case, still transferring its data to the nearest ES may lead to huge amount of transmission delay and unbearable waiting time of its task. In this situation, uploading the data from V_j to the cloud directly and removing such points in $G(v, E)$ can improve the overall layout. On the other hand, the RSU where a ES places should transfer its data to the ES.

Define D_j^m to represent whether V_j should transfer its data to ES_m in $G(v, E)$ which is calculated by

$$D_j^m = \begin{cases} 1, & \theta_j^m \leq \theta_{\max}, \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where θ_j^m is the length of shortest path from V_j to ES_m in $G(v, E)$ and θ_{\max} is the pre-set upper limit of path length. If V_j is not accessible to ES_m in $G(v, E)$, the value of θ_j^m is $+\infty$.

Therefore, the total number of the ESs that V_j could reach, which is marked as U_j , is calculated by

$$U_j = \sum_{m=1}^k D_j^m. \tag{4}$$

U_j is zero means that data acquired by V_j should be uploaded to the cloud directly, instead of being sent to any ES. What's more, V_j is supposed to be deleted from $G(v, E)$. The new graph without such points is marked as $G(v - v', E')$, where $|v - v'| = N$. Therefore, the RSUs in the new collection, which is denoted as $v - v'$, are indexed by $\{V_1, V_2, \dots, V_N\}$.

ESs are set up to preprocess data locally in order to reduce the pressure of cloud and the transmission energy consumption. As a result, the coverage rate of the ESs, i.e., C , can be measured from the proportion of data sent to ESs. C is the ratio of the amount of data sent to ESs to all collected data in unit time. In other words, it is the ratio of the amount of total data in $G(v, E)$ to that in $G(v - v', E')$ in unit time. Therefore, C is calculated by

$$C = \frac{\sum_{V_j \in v-v'} a_j}{\sum_{V_j \in v} a_j}, \tag{5}$$

where a_j is the amount of data collected by V_j per unit time.

3.3 Load Variance Model

In IoV, every RSU transfers its data to the nearest ES to reduce the waiting time of the tasks. If the number of the closest ES is more than one, RSU can transfer its data to either of them. Define L_j^m to represent the object of data transmission for V_j in $G(v - v', E')$ which is calculated by

$$L_j^m = \begin{cases} 1, & V_j \text{ transfers its data to } ES_m, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The workload of the m -th ES denoted as ES_m is the sum of the amount of data it receives per unit time, which is calculated by

$$g(ES_m) = \sum_{j=1}^N L_j^m a_j, \quad (7)$$

where a_j is the amount of data transferred by V_j in unit time and L_j^m represents whether ES_m analyzes the data from V_j .

The average workload of the ESs, i.e., \bar{g} , is one k -th of the total workload which is equal to the total amount of data received. It is calculated by

$$\bar{g} = \frac{1}{k} \sum_{j=1}^k g(ES_j). \quad (8)$$

The ideal situation of the workload is that every ES has the same workload, which will improve the reliability of the layout. Therefore, the load variance of the ESs, i.e., A , is calculated by

$$A = \frac{1}{k} \sum_{j=1}^k (g(ES_j) - \bar{g})^2. \quad (9)$$

3.4 Waiting Time Model

The total number of tasks that ES_m needs to solve is defined as p_m , which is calculated by

$$p_m = \sum_{j=1}^N L_j^m. \quad (10)$$

When handling the tasks, ES takes the principle that first come first served. It would spend all of its resource on dealing with the task that arrives first. A task will enter queuing sequence and wait if the ES is working when it comes. Let V_h^m be the RSU whose task is the h -th solved task on ES_m . The waiting time is the total amount of time it takes from the beginning of data transmission to the start of data analyzing. When the server is idle, the waiting time is the time required for data transmission; otherwise it is the sum of the waiting time and processing time of the previous task. Define t_e as the time for transmitting the data to the ESs. It is calculated by

$$t_0 = \frac{a_h \cdot \theta_h}{c}. \quad (11)$$

Therefore, the waiting time of the task which is the h -th solved in ES_m is calculated by

$$te = \begin{cases} t_0, & h = 1, \\ \text{Max} \left\{ te + \frac{a_{h-1}}{\lambda}, t_0 \right\}, & h \geq 2, \end{cases} \quad (12)$$

where a_h , λ , θ_h , and c are the scale of data V_h^m transfers in unit time, the scale of data a single ES can process in unit time, the length of the shortest path from V_h^m to ES_m in $G(v - v', E')$, and the rate of data transmission among the RSUs, respectively.

The total waiting time on ES_m is the sum of the waiting time of the tasks processed on ES_m . Therefore, the total waiting time on ES_m is calculated by

$$f(ES_m) = \sum_{h=1}^{p_m} te. \quad (13)$$

The total waiting time of the tasks is the sum of the waiting time on the ESs. As a result, the average waiting time of the tasks is calculated by

$$T = \frac{1}{N} \sum_{j=1}^k f(ES_j). \quad (14)$$

3.5 Problem Formulation

Our goal is to increase the coverage rate of the ES, reduce the average waiting time, and equilibrate the pressure of data processing, which will contribute to ensuring the stability and improving the performance of the IoV system. The placement problem can be defined as a multi-objective optimization problem, which is written as

$$\text{Max}C, \quad \text{Min}A, \quad \text{Min}T, \quad (15)$$

with the constraint

$$C \geq C_{\min}, \quad (16)$$

where C_{\min} is the minimum coverage for ES. Equation (16) sets the lower limit of coverage and ensures that ES can cover most tasks, thereby enhancing the rationality of the obtained scheme. The meanings of C , A , and T are shown in (5), (9), and (14), respectively.

4 An NPGA-II-Based Multi-objective Strategy for Edge Server Placement

In this section, the problem is illustrated with binary encoding. Then, NPGA-II is applied to search the proper schemes. Elitist preservation strategy is also used to avoid the loss of optimum solution. Finally, Roulette algorithm is used to accelerate the convergence.

4.1 Encoding Strategy

Binary encoding is applied to represent the edge server locations in our method. In $G(v, E)$, the set of the RSUs is denoted as $V = \{V_1, V_2, \dots, V_M\}$. Since a RSU can place at most an ES, a binary digit can be adopted to represent whether there is an ES on the RSU. Therefore, a M -bit binary number i could represent one kind of placement strategy of ESs since the placement problem is discrete. The meaning of a_j^i which is the j -th bit in i is represented as

$$a_j^i = \begin{cases} 1, & \text{if there is an ES placing on } V_j, \\ 0, & \text{if there is not an ES placing on } V_j. \end{cases} \quad (17)$$

Since the total scale of the ESs is k , a_j^i should obey the following constraint:

$$\sum_{j=1}^M a_j^i = k. \quad (18)$$

4.2 Edge Server Placement Scheme Generation Based on NPGA-II

Fitness Functions and Constraints In this section, to measure the rationality of the layout, fitness functions are set up. They consist of three functions: (5), (9), and (14). They are coverage rate, workload variance, and the average wait time, respectively. By adopting NPGA-II, all these functions will be considered during the development process. The final solution found by our method must follow the constraints mentioned in (16).

Initialization At this stage, the population size Q , the maximum evolution I , and the mutation probability byc are determined, so are other parameters. Q should be an even number since the schemes are divided into several pairs in the following stages. First, Q M bit binary numbers are randomly generated as the

initial population of evolution. These Q binary numbers represent the ES position of the individuals in R . The binary numbers constitute the first generation of evolution.

Selection The scale of possible schemes is C_M^k since k RSUs are selected from M RSUs to place the ESs. As a result, as the scale of the RSUs increases, the solution space will expand at an intolerable speed. To find the global optimal solution, we are supposed to ensure that the distance between the found solutions is as far as possible. Since NPGA-II is suitable for solving this kind of multi-objective optimization problem, it is adopted in the selection section to obtain the appropriate schemes.

Let P be the set of final individuals of the previous generation. In the first generation of evolution, P consisted of the initial population that was previously generated. This stage will select Q schemes from P to generate a new scheme. The collection of selected schemes is defined as R . Championships are applied to select the proper schemes to compose the next generation. The champion level of the scheme i , expressed as $rank[i]$, is the scheme size P it controls. Therefore, $rank[i]$ is calculated by

$$rank[i] = |\{i' | i > i'\}|. \quad (19)$$

Each time, the champion will be selected by two individuals of P . Plans with higher championship levels will win the championship and be placed in R . Otherwise, the degree of congestion in P will be considered.

Let d_m^n be the Hamming distance between the plan i_m and the plan i_n . The value of d_m^n is calculated by

$$d_m^n = \sum_{k=1}^M (a_k^{i_m} + a_k^{i_n}) \% 2. \quad (20)$$

Define shared function $sh(d_m^n)$ to represent the degree of crowding between scheme i_m and scheme i_n which is calculated by

$$sh(d_m^n) = \begin{cases} 1 - \frac{d_m^n}{\sigma}, & d_m^n < \sigma, \\ 0, & \text{otherwise,} \end{cases} \quad (21)$$

where σ is the niche radius that was set up before. It is the expectation of the minimum distance between any two solutions in real Pareto solution collection.

The schemes in R are indexed by $\{i_1, i_2, \dots, i_{|R|}\}$. The niche count of i_m (represented as NC_m) is the sum of the schemes' shared functions. Therefore, NC_m is calculated by

$$NC_m = \sum_{n=1}^{|R|} sh(d_m^n). \quad (22)$$

The solution with smaller niche count is the ultimate winner and selected into the next generation. If there are two scenarios with the same championship level and niche, then the winner should be both of them. The elitist preservation strategy is executed when R is empty. The scheme with the highest championship level will be selected as the first individual of R to avoid losing high-quality genes. The selection will be repeated until the scale of R is M .

Crossover and Mutation After selecting the new individuals for the next generation, the scheme in R is divided into $\frac{Q}{2}$ pairs to generate the next generation of individuals. The first step in this phase is crossover. The input is a pair of R schemes, that is, i_p and i_q . The output of the algorithm is two new schemes, namely, i_c and i_d . The key idea of the algorithm is to retain the features of i_p and i_q while exchanging some parts of i_p and i_q . Suppose a_n^m is the n -th bit of i_m binary encoding. To preserve the characteristics, if a_n^p is equal to a_n^q , then a_n^c and a_n^d are the values of a_n^p . Otherwise, half of the bits in i_c binary encoding come from i_p . Half of them come from i_q . The same goes for the binary encoding of i_d .

In the mutation part, binary coding is slightly changed to avoid premature convergence. There may be two bits in a binary code, which have different values, swapping their values with each other and the corresponding quantum code. The probability of a mutation is set to byc . At this stage, the mutation occurs at most once.

Roulette Assessment function AF is set to estimate the rationality of a layout. A ideal layout is sure to have a high coverage rate. Thus, the degree of C is added specially to support such layouts. The assessment function is calculated by

$$AF = \frac{\omega \times A \times co + (1 - \omega) \times T}{C^4}, \quad (23)$$

where ω is the evaluation coefficient that determined before, co is set to make $A \times co$ and T in the same order of magnitude, and C , A , and T are defined in (5), (9), and (14), respectively. The smaller AF , the better the layout.

By applying AF , the best layout so far, i.e., i_{best} , is determined. The is also applied in this part. Therefore, the first one chosen to be the next generation is i_{best} .

The probability of each individual being chosen as the next generation is proportional to the value of AF . The smaller AF is, the more likely it is to be selected.

To increase the diversity of quantum encodings, mutation operation is carried out after the roulette. Two bits in the binary encoding may swap with each other during the mutation. The probability of exchange for each bit is equal. The swap happens at most once in the mutation stage.

Method Overview The aim of the algorithm is to find a proper scheme for ES placement. Binary encoding is used to represent the ES locations. By using NPGA-II, excellent individuals are chosen and put in R (Lines 6 and 7). Roulette algorithm is used to accelerate the convergence (Lines 8 and 9). Crossover and mutation are

Algorithm 1 Strategy for Edge Server Placement

Input: Q, I, θ_0, σ
Output: i_{best}

```

initialization( $P$ )
2:  $ge \leftarrow 0$ 
   while  $ge < I$  do
4:    $R \leftarrow \emptyset$ 
     while  $|R| < Q$  do
6:       if  $ge < I - 10$  then
           Selection
8:       else
           Roulette
10:      end if
     end while
12:     $j \leftarrow 1$ 
     while  $j \leq \frac{Q}{2}$  do
14:       Crossover( $i_j, i_{Q-j+1}$ )
       Mutation( $i_j, i_{Q-j+1}$ )
16:       $j \leftarrow j + 1$ 
     end while
18:     $P \leftarrow R$ 
     $ge \leftarrow ge + 1$ 
20: end while
    find  $i_{\text{best}}$  in  $P$  according to the assessment function shown in (23)
22: return  $i_{\text{best}}$ 

```

carried out successively to maintain the diversity of schemes (Lines 11 to 15). The best layout so far, i.e., i_{best} , is found and output in the final step (Lines 21 and 22).

5 Experimental Evaluation

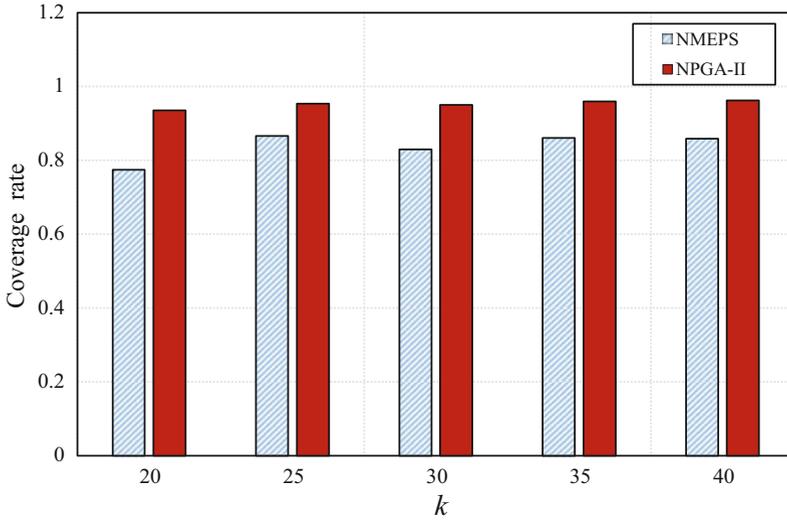
In this section, the service performance of NMEPS is evaluated by the real vehicle big data. NMEPS is compared with NPGA-II which selects the next generation based on the number of niche in this section. In NPGA-II, RSUs with distances greater than θ_{max} transmit their data to the cloud, where θ_{max} is the predetermined maximum distance for data transmission. The parameter settings in this experiment are shown in the Table 1

5.1 Comparison on Coverage Rate

Coverage is the ratio of the size of the data processed on the ES to the size of the data collected by the RSUs.

Table 1 Parameter settings

Parameter	Value	Parameter	Value
M	436	k	30
Q / I	200	θ_{\max}	7
r_{\max}	1500	σ	7
λ	200,000	c	1,000,000
byc	0.1	co	10^{-13}
C_{\min}	0.7	θ_0	0.05π
ω	0.5		

**Fig. 2** Comparison on coverage rate by NMEPS and NPGA-II at different scales of the edge servers

As shown in Fig. 2, the coverage rates of both algorithms are greater than the minimum coverage rate determined before. The coverage of NPGA-II is much higher than the coverage rate of NMEPS, while the coverage rate of NMEPS is also acceptable in most cases. Therefore, when the scheme is obtained through NMEPS instead of NPGA-II, the coverage rate will not be greatly reduced.

5.2 Comparison on Workload Variance

The workload variance is applied to indicate the degree of dispersion of workload in the scheme. Since the reliability of this solution depends on the ES with the shortest service life, solutions with smaller workload variances are more likely to work longer and have higher reliability. Figure 3 shows that the workload variance of NMEPS is much lower than that of NPGA-II, which is very good for ensuring

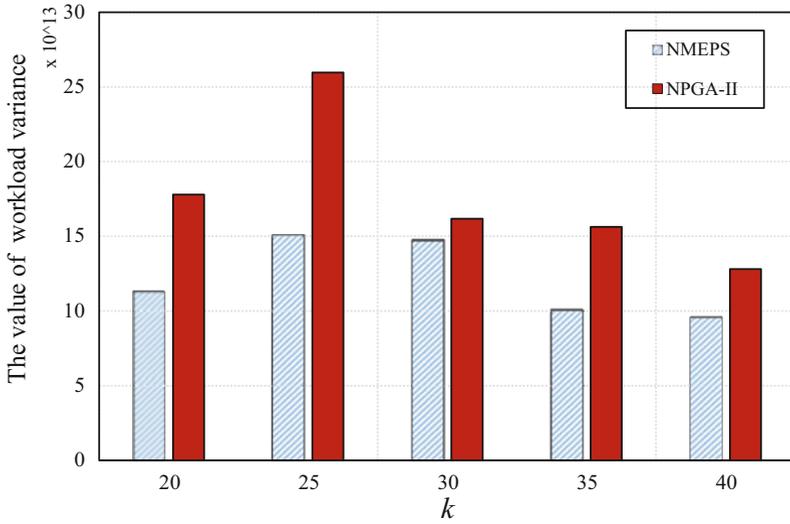


Fig. 3 Comparison on workload variance by NMEPS and NPGA-II at different scales of the edge servers

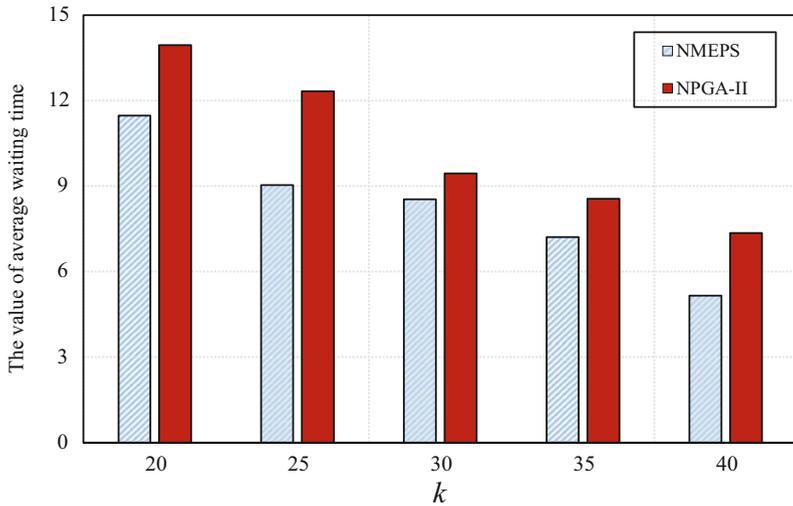


Fig. 4 Comparison on average waiting time by NMEPS and NPGA-II at different scales of the edge servers

the reliability of the layout. The low workload variance also means that the ES is less likely to fail and the available work time is longer.

5.3 Comparison on Average Waiting Time

The waiting time is defined as the sum of the time required for data transmission and the time spent waiting for the ES. Because the ES is limited by resources, when the ES is busy, tasks that arrive will wait to be processed. Since NMEPS applies roulette in the evolution process, plans with high value AF are more likely to survive and enter the next generation. As a result, when the scheme is obtained using NMEPS, the average waiting time is reduced and the efficiency of the layout is improved.

6 Conclusion

In the IoV system, the location of the ES has a great impact on the performance of the ES. To obtain a suitable ES placement scheme, NPGA-II-based NMEPS was proposed. By applying appropriate evaluation functions, you can ensure that the schemes discovered by NMEPS have higher coverage and lower workload variance. Compared with NPGA-II, because NMEPS uses a roulette algorithm in the evolution process, the scheme obtained by NMEPS shows better performance in the experimental part. In future work, we will change the parameters in the algorithm to achieve a balance between NPGA-II and roulette. We will also work to find a suitable population as the initial population to further improve the service performance of the final plan.

Acknowledgments This work is supported by the National Key R&D Program of China under Grant 2019YFE0190500, the National Natural Science Foundation of China under grant no.61702277 and no.61872219, the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund and Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAET).

References

1. N.A. Bakar, M.K.M. Desa, Optimal placement of tesc in transmission network using sensitivity based method for multi-objective optimization, in *2017 IEEE Conference on Energy Conversion (CENCON)* (2017), pp. 201–206
2. L. Chen, J. Xu, S. Ren, P. Zhou, Spatio-temporal edge service placement: a bandit learning approach. *IEEE Trans. Wirel. Commun.* **17**(12), 8388–8401 (2018)
3. M. De Donno, K. Tange, N. Dragoni, Foundations and evolution of modern computing paradigms: cloud, IoT, edge, and fog. *IEEE Access* **7**, 150936–150948 (2019)
4. I. Hadžić, Y. Abe, H.C. Woithe, Server placement and selection for edge computing in the epc. *IEEE Trans. Serv. Comput.* **12**(5), 671–684 (2019)
5. P. Hao, L. Hu, J. Jiang, J. Hu, X. Che, Mobile edge provision with flexible deployment. *IEEE Trans. Serv. Comput.* **12**(5), 750–761 (2019)
6. Q. He, G. Cui, X. Zhang, F. Chen, S. Deng, H. Jin, Y. Li, Y. Yang, A game-theoretical approach for user allocation in edge computing environment. *IEEE Trans. Parallel Distrib. Syst.* **31**(3), 515–529 (2020)

7. C. Li, L. Toni, J. Zou, H. Xiong, P. Frossard, Qoe-driven mobile edge caching placement for adaptive video streaming. *IEEE Trans. Multimedia* **20**(4), 965–984 (2018)
8. L. Lin, X. Liao, H. Jin, P. Li, Computation offloading toward edge computing. *Proc. IEEE* **107**(8), 1584–1607 (2019)
9. J. Ren, G. Yu, Y. He, G.Y. Li, Collaborative cloud and edge computing for latency minimization. *IEEE Trans. Veh. Technol.* **68**(5), 5031–5044 (2019)
10. S.N. Shirazi, A. Gouglidis, A. Farshad, D. Hutchison, The extended cloud: review and analysis of mobile edge computing and fog from a security and resilience perspective. *IEEE J. Sel. Areas Commun.* **35**(11), 2586–2595 (2017)
11. P. Smet, B. Dhoedt, P. Simoens, Docker layer placement for on-demand provisioning of services on edge clouds. *IEEE Transactions on Network and Service Management*, **15**(3), 1161–1174 (2018)
12. X. Xu, Y. Xue, L. Qi, X. Zhang, S. Wan, W. Dou, V. Chang, Load-aware edge server placement for mobile edge computing in 5g networks, in *International Conference on Service-Oriented Computing* (Springer, 2019), pp. 494–507
13. Q. Yan, X. Tang, Q. Chen, M. Cheng, Placement delivery array design through strong edge coloring of bipartite graphs. *IEEE Commun. Lett.* **22**(2), 236–239 (2018)
14. T.T.T. Zan, P. Gupta, M. Wang, J. Dauwels, A. Ukil, Multi-objective optimal sensor placement for low-pressure gas distribution networks. *IEEE Sens. J.* **18**(16), 6660–6668 (2018)
15. L. Zhao, J. Liu, Optimal placement of virtual machines for supporting multiple applications in mobile edge networks. *IEEE Trans. Veh. Technol.* **67**(7), 6533–6545 (2018)

Automatic Mapping of a Physical Model into a Conceptual Model for a NoSQL Database



Fatma Abdelhedi, Amal Ait Brahim, Rabah Tighilt Ferhat, and Gilles Zurfluh

1 Introduction

Big Data have attracted a great deal of attention in recent years, thanks to the huge amount of data managed, the types of data supported, and the speed at which these data are collected and analyzed. This has impacted the tools required to store Big Data, and new kinds of data management tools, that is, not only structured query language (NoSQL) systems, have arisen [1]. Compared to existing database management systems (DBMSs), NoSQL systems are generally accepted to support greater data volume and to ensure faster data access, undeniable flexibility, and scalability [2].

One of the NoSQL key features is that databases can be schema-less. This means, in a table, meanwhile the row is inserted, the attributes names and types are specified. This property offers an undeniable flexibility that facilitates the data model evolution and allows end-users to add new information without the need of database administrator; but, at the same time, it makes the database manipulation more difficult. Indeed, even in Big Data context, the user still needs a data model that offers a visibility of how data are structured in the database (table's name, attributes names and types, relationships, etc.). In practice, the developer who has created the database is also in charge of writing queries. Thus, he/she already knows how data are stored and related in the database, so he/she can easily express his requests.

F. Abdelhedi

Toulouse Institute of Computer Science Research (IRIT), Toulouse Capitole University,
Toulouse, France

CBI2 – TRIMANE, Paris, France

A. Ait Brahim · R. Tighilt Ferhat (✉) · G. Zurfluh

CBI2 – TRIMANE, Paris, France

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_40

557

However, this solution cannot be applied to all cases; for instance, the developer who is asked for doing the application maintenance does not know the data model. It is the same for a decision maker who needs to query a database while he/she was not involved in its creation.

On one hand, NoSQL systems have proven their efficiency to handle Big Data. On the other hand, the needs of a NoSQL database model remain up to date. Therefore, we are convinced that it is important to provide to users (developers and decision-makers) two data models describing the database: (1) a physical model that describes the internal organization of data and allows to express queries and (2) a conceptual model that provides a high level of abstraction and a semantic knowledge element close to human comprehension, which guarantees efficient data management [3].

In previous works, we have proposed a process for extracting a physical model starting from a NoSQL database. In this paper, we aim to propose an extension of this work by transforming the physical model (already obtained) into a conceptual model; a reverse engineering process will be used for this.

The remainder of the paper is structured as follows. Section 2 motivates our work using an example in the healthcare field. Section 3 reviews previous work. Section 4 describes our reverse engineering process. Section 5 details our experiments, compares our solution against those presented in Sect. 3, and validates our solution. Finally, Sect. 6 concludes the paper and announces future work.

2 Illustrative Example

To motivate and illustrate our work, we have used a case study in the healthcare field. This case study concerns international scientific programs for monitoring patients suffering from serious diseases. The main goal of this program is (1) to collect data about diseases development over time, (2) to study interactions between different diseases, and (3) to evaluate the short- and medium-term effects of their treatments. The medical program can last up to 3 years. Data collected from establishments involved in this kind of program have the features of Big Data (the 3 V): Volume: the amount of data collected from all the establishments in 3 years can reach several terabytes. Variety: data created while monitoring patients come in different types; it could be (1) structured as the patient's vital signs (respiratory rate, blood pressure, etc.), (2) semi-structured document such as the package leaflets of medicinal products, and (3) unstructured such as consultation summaries, paper prescriptions, and radiology reports. Velocity: some data are produced in continuous way by sensors; it needs a [near] real-time process because it could be integrated into a time-sensitive processes (e.g., some measurements, like temperature, require an emergency medical treatment if they cross a given threshold).

In these programs, one of the benefits of using NoSQL databases is that the evolution of the data (and the model) is fluent. In order to follow the evolution of the pathology, information is entered regularly for a cohort of patients. But the situation

of a patient can evolve rapidly which needs the recording of new information. Thus, few months later, each patient will have his own information, and that is how data will evolve over time. Therefore, the data model (1) differs from one patient to another and (2) evolves in unpredictable way over time.

As mentioned before, these kind of systems operate on schema-less data model enabling developers to quickly and easily incorporate new data into their applications without rewriting tables. Nevertheless, there is still a need for a conceptual model to know how data are structured and related in the database; this is particularly necessary to write declarative queries where tables and columns names are specified [4].

In our view, it is important to have a precise and automatic solution that guides and facilitates the database model extraction task within NoSQL systems. For this, we propose the *ToConceptualModel* process presented in Sect. 4 that extracts a conceptual model of a NoSQL database. This model is expressed using a Unified Modeling Language (UML) class diagram.

3 Related Work

The problem of extracting the data model from schema-less NoSQL databases has been the subject of several research works. Most of these works focus on the physical level [3, 5–10]. In this context, we have proposed a process to extract a document-oriented database physical model [11]. This process applies a sequence of transformations formalized with the Query/Views/Transformation (QVT) standard proposed by the Object Management Group (OMG). What is original in our solution is that it takes into account the links between different collections.

However, we should highlight that only few works [12–14] have addressed the extraction of a NoSQL database conceptual model. In the study cited in [12], the authors propose an extraction process of a conceptual model for a graph-oriented NoSQL database (Neo4J). In this particular type of NoSQL databases, the database contains nodes (objects) and binary links between them. The proposed process takes as input the insertion requests of objects and links and then returns an Entity/Association model. This process is based on model-driven architecture (MDA) and successively applies two transformations. The first is to build a graph (Nodes + Edges) from the Neo4j query code. The second consists of extracting an Entity/Association model from this graph by transforming the nodes with the same label into entities and the edges into associations. These works are specific to graph-oriented NoSQL databases generally used to manage strongly linked data such as those from social networks.

Authors in [13] propose a process to extract a conceptual model (UML class diagram) from a JavaScript Object Notation (JSON) document. This process consists of two steps. The first step extracts a physical model in JSON format. The second step generates the UML class diagram by transforming the physical model into a root class (RC), the primitive fields (Number, String, and Boolean)

into attributes of RC, and the structured fields into component classes linked to RC by composition links. Thus, this work only considers the composition links and ignores other kinds of links (e.g., association and aggregation links).

The process proposed in [14] deals with the mapping of a document-oriented database into a conceptual model. It consists of a set of entities with one or more versions. This work considers the two types of links: binary association and composition. Binary association and composition links are respectively extracted using the reference and structured fields. This solution does not consider other links that are usually used like n-ary, generalization, and aggregation links.

In Table 1, we summarize the three previous works according to three factors: modeling levels (physical and/or conceptual), NoSQL system type, and links types.

Regarding the state of the art, the existing solutions have the advantage to start from the conceptual level, but they do not consider all the UML class diagram features that we need for our medical use case. Indeed, the process in [12] concerns only graph-oriented systems that, unlike document-oriented databases, do not allow to express structured attributes and composition links. On the other hand, the solution cited in [13] starts from a document-oriented database but do not consider aggregation, generalization, and association links that are used to link data in our case study. As cited in [13], authors in [14] use a document-oriented database, but do not take into account the generalization and aggregation links, association classes, and also n-ary association links that are the most used in the medical application.

The main purpose of our work is to complete these solutions. For this, we have proposed an automatic approach that transforms a document-oriented physical model into a UML class diagram. This automatic process addresses several kinds of links: association classes, binary and n-ary association links, and also the generalization, composition, and aggregation links.

4 Reverse Engineering Process

Our work aims to provide users with models to manipulate NoSQL databases. Two models are proposed: (1) the physical model to write queries on this database and application code and (2) the conceptual model to give the meaning of the data contained in the database. When data structures are complex, these two models are

Table 1 Comparative table of extraction works of conceptual model from schema-less NoSQL databases

	Modeling level		NoSQL system type		Link types		
	Physical	Conceptual	Graph	Document	Binary association	Composition	Generalization
[12]	X	X	X		X		X
[13]	X	X		X		X	
[14]	X	X		X	X	X	

essential to enable users (usually, developers and decision-makers) to understand and manipulate data independently.

As part of this work, we proposed mechanisms for discovering a physical model from a NoSQL database in a previous paper. The current paper completes the latter and focuses on the transformation of the physical model into a conceptual model represented by using a UML class diagrams (red frame in Fig. 1) and which provides users with the semantics of the data.

Note that we limit our study to document-oriented NoSQL databases that are the most complete to express links between objects (use of referenced and nested data).

We propose the *ToConceptualModel* process which applies a set of transformations ensuring the passage of a NoSQL physical model toward a UML class diagram.

In the following sections, we detail the components of the *ToConceptualModel* process by specifying the three elements: (a) the source, (b) the target, and (c) the transformation algorithms.

4.1 Source: Physical Model

The physical model is produced by the *ToPhysicalModel* process (shown in Fig. 1). In this paper, it is the source of the *ToConceptualModel* process that we will study here. The physical model is defined as a pair (N, CL) , where:

- N is the physical model name,
- $CL = \{cl_1, \dots, cl_n\}$ is a set of collections.
 - $\forall i \in [1..n]$, a collection $cl_i \in CL$ is defined as a pair (N, F) , where:
 - $cl_i.N$ is the collection name,
 - $cl_i.F = AF \cup SF$ is a set of fields, where:
 - $AF = \{af_1, \dots, af_m\}$ is a set of atomic fields. $\forall j \in [1..m]$, an atomic field $af_j \in AF$ is defined as a tuple (N, T, M) , where:

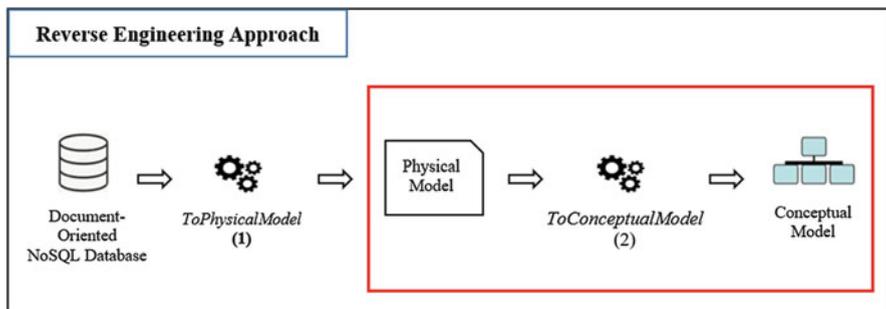


Fig. 1 Overview of *ToConceptualModel* process

- $af_j.N$ is the af_j name,
 - $af_j.T$ is the af_j type; it is one of the standards data types such as Integer, String, Boolean, . . . ,
 - $af_j.M$ is a boolean which indicates whether af_j is multivalued or not.
- $SF = \{sf_1, \dots, sf_l\}$ is a set of structured fields. $\forall k \in [1..l]$, a structured field $sf_k \in SF$ is defined as a tuple (N, F', M) , where:
 - $sf_k.N$ is the sf_k name,
 - $sf_k.F' = AF' \cup SF'$ is a set of fields that compose the structured field sf_k (see above),
 - $sf_k.M$ is a boolean which indicates whether sf_k is multivalued or not.

To express a link between two collections, we used a field called DBRef, which is unstandard proposed by MongoDB [15]. A DBRef field is a special case of a structured field (N, F', M) , where: N is the link name; F' contains two atomic fields: ***\$id: ObjectId*** which corresponds to the identifier of the referenced document and ***\$Ref: String*** which corresponds to the name of the collection that contains the referenced document; M indicates whether the link is monovalued or multivalued. We have extended the DBRef syntax to take into account n-ary links and association classes. In this new syntax, F' can contain several pairs (***\$id: ObjectId, \$Ref: String***) and possibly other fields.

To create a generalization link between collections, we propose using a DBSub field in the sub-collection. DBSub is a special case of structured field where $N = \text{Sub}$; F' contains two atomic fields: ***\$id: ObjectId*** which identifies the generic document and ***\$Sub: String*** corresponds to the super-collection name; $M = 0$. To express an aggregation link between collections, we suggest using a DBAgg field in the aggregate collection. DBAgg is a special case of structured field (N, F', M) where $N = \text{Agg}$; F' contains two atomic fields, ***\$id: ObjectId*** which identifies the part document (aggregated), and ***\$Agg: String*** corresponds to the name of the part collection.

4.2 Target: Conceptual Model

A UML Class Diagram (CD) is defined as a tuple (N, C, L) , where:

- N is the CD name.
- $C = \{c_1, \dots, c_n\}$ is a set of classes.
- $L = \text{AL} \cup \text{CL} \cup \text{AGL} \cup \text{GL}$ is a set of links.

Classes

$\forall i \in [1..n]$, a class $c_i \in C$ is defined as a pair (N, A) , where:

- $c_i.N$ is the class name,
- $c_i.A = \text{AA} \cup \text{SA}$ is a set of attributes, where:

- $AA = \{aa_1, \dots, aa_m\}$ is a set of atomic attributes.
 $\forall j \in [1..m]$, an atomic attribute $aa_j \in AA$ is defined as a tuple (N, T, M) , where:
 - $aa_j.N$ is the aa_j name,
 - $aa_j.T$ is the aa_j type; T can have the value: String, Integer, Boolean...
 - $aa_j.M$ is a boolean which indicates whether
 - aa_j is multivalued or not.
- $SA = \{sa_1, \dots, sa_l\}$ is a set of structured attributes.
 $\forall k \in [1..l]$, a structured attribute $sa_k \in SA$ is defined as a tuple (N, A', M) , where:
 - $sa_k.N$ is the sa_k name,
 - $sa_k.A' = AA' \cup SA'$ is a set of attributes that compose sa_k (see above),
 - $sa_k.M$ is a boolean which indicates whether sa_k is multivalued or not.

Links

- $AL = \{al_1, \dots, al_m\}$ is a set of association links.
 $\forall i \in [1..m]$, an association link $al_i \in AL$ is defined as a tuple (N, RC, A) , where:
 - $al_i.N$ is the al_i name,
 - $al_i.RC = \{rc_1, \dots, rc_f\}$ a set of related collections with degree $f \geq 2$. $\forall j \in [1..f]$, rc_j is defined as a pair (c, cr) , where:
 - $rc_j.c$ is the related class name,
 - $rc_j.cr$ is the multiplicity corresponding to c .
 - $al_i.A = AA \cup SA$ is a set of attributes of al_i (see above). Note that if $al_i.A \neq \emptyset$ then al_i is an association class.
- $CL = \{cl_1, \dots, cl_m\}$ is a set of composition links.
 $\forall i \in [1..m]$, a composition link $cl_i \in CL$ is defined as a pair $(rc^{composite}, rc^{component})$ where :
 - $cl_i.rc^{composite}$ is a pair defining the composite class; it is in the form of (c, cr) , where:
 - $rc^{composite}.c$ is the composite class name.
 - $rc^{composite}.cr$ is the multiplicity corresponding to the composite class. This multiplicity generally contains the value 0..1, 1..1 or 1 for the contracted form.
 - $cl_i.rc^{component}$ is a pair defining the component class; it is in the form of (c, cr) , where:
 - $rc^{component}.c$ is the component class name.
 - $rc^{component}.cr$ is the multiplicity corresponding to the component class.

- $AGL = \{agl_1, \dots, agl_m\}$ is a set of aggregation links.
 - $\forall i \in [1..m]$, an aggregation link $agl_i \in AGL$ is defined as a pair $(rc^{aggregate}, rc^{part})$, where:
 - $agl_i.rc^{aggregate}$ is a pair defining the aggregate class; it is in the form of (c, cr) , where:
 - $rc^{aggregate}.c$ is the aggregate class name,
 - $rc^{aggregate}.cr$ is the multiplicity corresponding to the aggregate class,
 - $agl_i.rc^{part}$ is a pair defining the part class; it is in the form of (c, cr) , where:
 - $rc^{part}.c$ is the part class name,
 - $rc^{part}.cr$ is the multiplicity corresponding to the part class,
- $GL = \{gl_1, \dots, gl_m\}$ is a set of generalization links.
 - $\forall i \in [1..m]$, a generalization link $gl_i \in LH$ is defined as a pair (sc, SSC) , where:
 - $gl_i.sc$ is the super-class name,
 - $gl_i.SBC = \{sbc_1, \dots, sbc_k\}$, where: $\forall j \in [1..k]$, with $k \geq 1$, sbc_j is a sub-class.

4.3 Transformation Algorithms

The mapping from the physical model to the conceptual model is ensured by applying six transformation algorithms: $TA_{Collection}$, $TA_{AtomicField}$, $TA_{StructuredField}$, TA_{DBRef} , TA_{DBSub} and TA_{DBAgg} .

- $TA_{Collection}$

This algorithm transforms a collection into a class. $TA_{Collection}$ possesses the following properties:

Input

$cl = (N, F)$: a collection defined by a name N and a set of fields $F = AF \cup SF$.

Output

$c = (N, A)$: a class defined by a name N and a set of attributes $A = AA \cup SA$.

Definition

$TA_{Collection}$

In: cl

Out: c

1Begin

2 $c.N = cl.N$

3 $c.A$ is generated by applying algorithms relating to the transformation of the collection fields.

4 $C = CU\ c // Add\ c\ to\ the\ set\ classes\ C$

5End

- $TA_{AtomicField}$

This algorithm transforms an atomic field into an atomic attribute. $TA_{AtomicField}$ possesses the following properties:

Input

$af = (N, T, M)$: an atomic field defined by a name N , a type T , and a boolean M .

Output

$aa = (N, T, M)$: an atomic attribute defined by a name N , a type T , and a boolean M .

Definition

TA *AtomicField*

In: af

Out: aa

1Begin

2 $aa.N = af.N$

3 $aa.T = af.T$

4 $aa.M = af.M$

5End

• **TA** *StructuredField*

This algorithm transforms a structured field which is not a DBRef. The result of this transformation can be either a composition link if the structured field consists of at least one DBRef field or a structured attribute otherwise. $TA_{StructuredField}$ possesses the following properties:

Input

$sf = (N, F', M)$: a structured field defined by a name N , a set of fields $F' = AF' \cup SF'$, and a boolean M . sf is declared in the collection c_0 .

Output

$cl = (rc^{composite}, rc^{component})$: a composition link defined by a composite class $rc^{composite}$ and a component class $rc^{component}$.

Or $sa = (N, A', M)$: a structured attribute defined by a name N , a set of attributes A' , and a boolean M .

Definition

TA *StructuredField*

In: sf

Out: cl or sa

1Begin

2 **If** $\exists dbref \in sf$. SF' **Then** // This is a composition link

3 $cl.rc^{composite}.c = c_0.N$

4 $cl.rc^{composite}.cr = 1..1$

5 $cl.rc^{component}.c = sf.N$

6 **If** $sf.M = 0$ **Then** // If sf is monovalued

7 $cl.rc^{component}.cr = 0..1$

8 **Else** // If sf is multivalued

9 $cl.rc^{component}.cr = 0..*$

10 **End If**

11 Get al by applying **TA** *DBRef* on $dbref$ // Create an association link between the component class and the class referenced in $dbref$

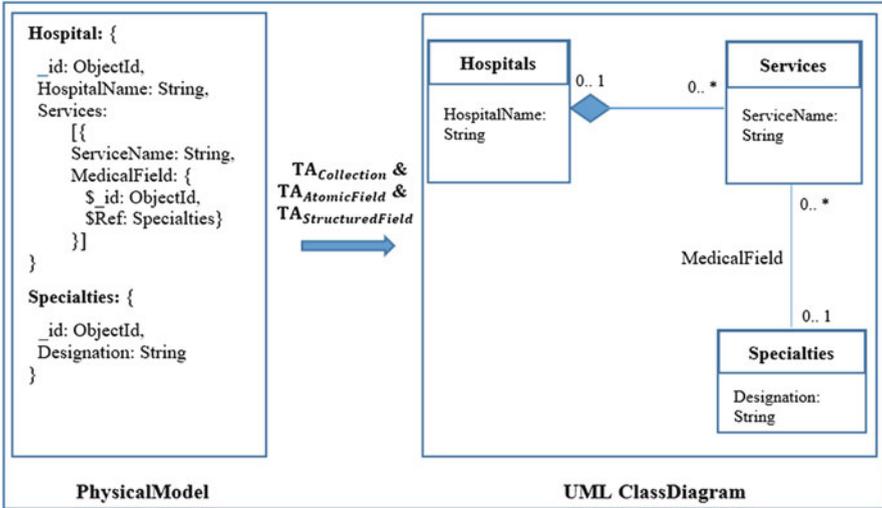


Fig. 2 Example of extracting a composition link

```

12 Else// This is a structured attribute
13     sa.N = sf.N
14     For every afi ∈ sf. AF', with i ∈ [1..m] do
15         Get aai by applying TAAtomicField
16         sa. AA' = sa. AA ∪ aai
17     End For
18     For every sfj ∈ sf. SF', with j ∈ [1..l] do
19         Get saj by applying TAStructuredField
20         sa. SA' = sa. SA ∪ saj
21     End For
22     sa.M = sf.M
23 End If
24 End

```

Example 1

In the example illustrated in Fig. 2, our process applies algorithms: *TA_{Collection}*, *TA_{AtomicField}* et *TA_{StructuredField}* as follows:

- The classes *Hospitals* and *Specialties* are obtained by applying *TA_{Collection}* on the collections *Hospitals* and *Specialties*.
- The atomic attributes *Hospital Name*, *Service Name*, and *Designation* are obtained by applying *TA_{AtomicField}*.
- The composition link between *Hospitals* and *Services* *TA_{StructuredField}*.

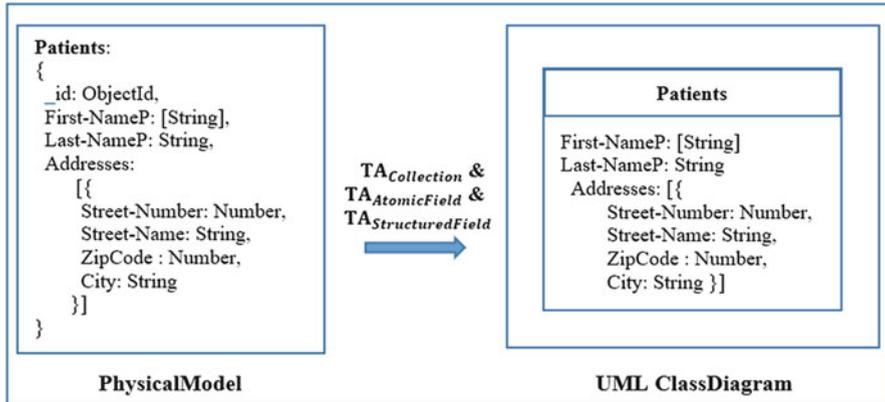


Fig. 3 Example of extracting a structured attribute

Example 2

In the example illustrated in Fig. 3, the structured attribute addresses in the collection patients is obtained by applying $TA_{StructuredField}$.

- TA_{DBRef}

This algorithm transforms a DBRef field into an association link. TA_{DBRef} possesses the following properties:

Input

$dbref = (N, F', M)$: a DBRef field defined by a name N , a set of fields F' (composed of n pairs ($\$id$: $ObjectId$, $\$Ref$: C_i) with $i \in [1..n]$) and possibly, m atomic fields and l structured fields), and a boolean M . $dbref$ is declared in the collection c_0 .

Output

$al = (N, RC, A)$: an association link defined by a name N , a set of related classes RC , and a set of attributes $A = AA \cup SA$.

Definition

TA_{DBRef}

In: $dbref$

Out: al

lBegin

2 $al.RC = \emptyset$

3 $al.A = \emptyset$

4 $al.N = dbref.N$

5 **If** $n = 1$ **Then**// It is a binary link

6 $rc_0.c = c_0.N$

7 $rc_1.c = c_1.N$

8 $rc_0.cr = 0..*$

9 **If** $dbref.M = 0$ **Then**// if $dbref$ is monovalued

10 $rc_1.cr = 0..1$

11 **else**// if $dbref$ is multivalued

12 $rc_1.cr = 0..*$

13 **End If**

```

14     al. RC = rc0 ∪ rc1
15 Else //n > 1 (It is an n-ary link)
16     rc0.c = c0.N
17     If dbref.M = 0 Then// if dbref is monovalued
18         rc0.cr = 0..1
19     Else// if dbref is multivalued
20         rc0.cr = 0..*
21     End If
22     al. RC = rc0
23     For i ∈ [1..n] do
24         rci.c = ci.N
25         rci.cr = 0..*
26     End For
27     End For
28 End If
29 For every afj ∈ dbref. AF', with j ∈ [1..m] do// Extract the
    atomic attributes of the association class
30     Get aaj by applying TA AtomicField
31     al. AA = al. AA ∪ aaj
32 End For
33 For every sfk ∈ dbref. SF', with k ∈ [1..l] do// Extract the
    structured attributes of the association class
34     Get sak by applying TA StructuredField
35     al. SA = al. SA ∪ sak
36 End For
37 al. A = al.AA ∪ al.SA
38End

```

Example 3

In the example illustrated in Fig. 4, the association link treating-doctors between patients and doctors is obtained by applying *TADBRef*.

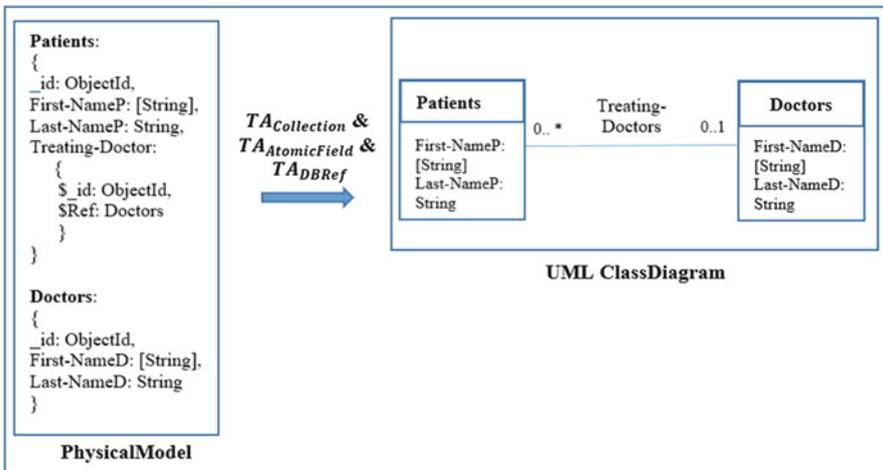


Fig. 4 Example of extracting an association link

Example 4

In the example illustrated in Fig. 5, the association class consultations between patients and doctors is obtained by applying TA_{DBRef} .

- TA_{DBSub}

This algorithm transforms a DBSub field into a generalization link. TA_{DBSub} possesses the following properties:

Input

$dbsub = (N, F', M)$: a DBSub field defined as a structured field whose $N = \text{Sub}$; F' consists of two atomic fields: $\$id: \text{ObjectId}$ and $\$Sub: \text{SCL}$; the value of M is 0. $dbsub$ is declared in the collection SbCl.

Output

$gl = (sc, SBC)$: a generalization link defined by a super-class sc and a set of sub-classes SBC .

Definition

TA_{DBSub}

In: $dbsub$

Out: gl

1Begin

2 $gl.sc = SCL$

3 $gl.SBC = gl.SBC \cup SbCl$

4End

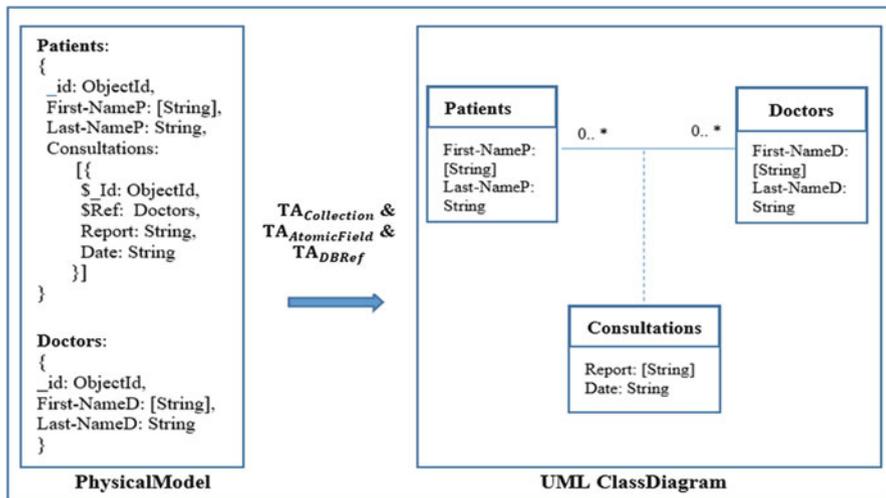
Example 5

Fig. 5 Example of extracting an association class

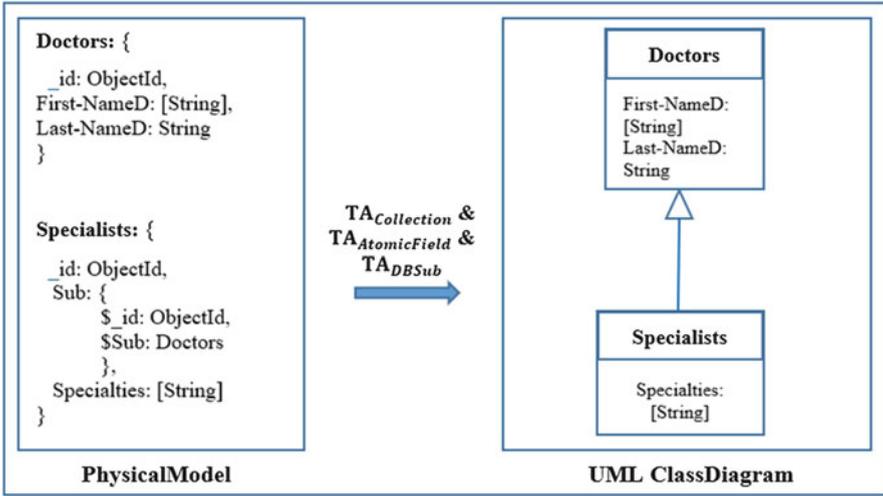


Fig. 6 Example of extracting a generalization link

In the example illustrated in Fig. 6, the generalization link between specialists and doctors is obtained by applying TA_{DBSub} .

- TA_{DBAgg}

This algorithm transforms a DBAgg field into an aggregation link. TA_{DBAgg} possesses the following properties:

Input

$dbagg = (N, F', M)$: a DBAgg field defined by $N = Agg$, a set of fields F' (consists of two atomic fields $\$id: ObjectId$ and $\$Agg: c_1$) and a boolean $M = 0$. $dbagg$ is declared in the part collection c_0 .

Output

$agl = (rc^{aggregate}, rc^{part})$: an aggregation link defined by an aggregate class $rc^{aggregate}$ and a part class rc^{part} .

Definition

TA_{DBAgg}

In: $dbagg$

Out: agl

1Begin

3 $agl.rc^{aggregate}.c = c_1.N$

4 $agl.rc^{part}.c = c_0.N$

5 $agl.rc^{part}.cr = 0..*$

6 **If** $dbagg.M = 0$ **Then**// if $dbagg$ is monovalued

7 $agl.rc^{aggregate}.cr = 0..1$

8 **Else**//if $dbagg$ is multivalued

9 $agl.rc^{aggregate}.cr = 0..*$

10 **End If**

14**End**

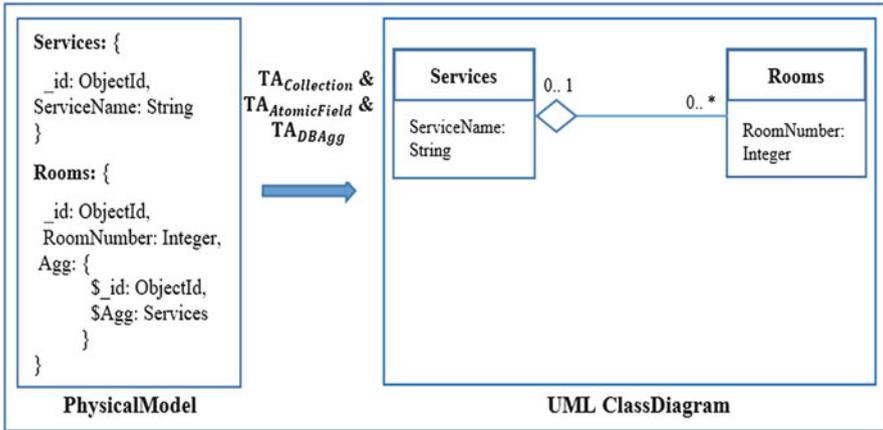


Fig. 7 Example of extracting an aggregation link

Example 6

In the example illustrated in Fig. 7, the aggregation link between services and rooms is obtained by applying TA_{DBAgg} .

5 Experiments and Comparison

5.1 Technical Environment

In this section, we describe the techniques used to implement the *ToConceptualModel* process. We used a technical environment suitable for modeling, meta-modeling, and model transformation. We used the Eclipse Modeling Framework (EMF) [16]. EMF provides a set of tools for introducing a model-driven development approach within the Eclipse environment. These tools provide three main features. The first is the definition of a meta-model representing the concepts handled by the user. The second is the creation of the models instantiating this meta-model and the third one is the transformation from model to model and from model to text. Among the tools provided by EMF, we used:

- Ecore: a meta-modeling language used to create our metamodels. Figures 2 and 3 illustrate the source and target Ecore meta-models used by *ToConceptualModel* process.
- XML Metadata Interchange (XMI): the XML-based standard that we use to create models.
- QVT (Query, View, Transformation): the OMG standard language for specifying model transformations. The choice of the QVT standard was based on criteria specific to our approach. Indeed, the transformation tool must be integrated into

the EMF environment so that it can be easily used with modeling and meta-modeling tools.

5.2 *Implantation of the ToConceptualModel Process*

ToConceptualModel process is expressed as a sequence of elementary steps that build the resulting model (UML class diagram) step by step from the source model (physical model).

Step1: we create a source and a target meta-model to represent the concepts handled by our process.

Step2: we build an instance of the source meta-model. For this, we use the standard based XML Metadata Interchange (XMI) format. This instance is shown in Fig. 8.

Step3: we implement the transformation algorithms by means of the QVT language provided within EMF.

Step4: we test the transformation by running the QVT script created in step 3. This script takes as input the source model built in step 2 (physical model) and returns as output a UML class diagram. The result is provided in the form of XMI file as shown in Fig. 9.

5.3 *Comparison*

The aim of this section is to compare our solution with the three works [12–14] presented in Sect. 3 and that have investigated the process of extracting a NoSQL database conceptual model. Starting from a graph-oriented NoSQL database, authors in [12] propose to extract an E/A model based on a set of mapping rules between the conceptual level and the physical one. Obviously, these rules are specific to graph-oriented systems used as a framework for managing complex data with many connections. This kind of NoSQL DBMS lacks ability to define structured attributes, association classes as well as n-ary, composition, and aggregation links that we need to use in our use case (cf. Sect. 2). The solution presented in [13] has the advantage to start from a document-oriented NoSQL database. The only type of links considered in this work is the composition link; other types are not taken into account. Other process in [14] focuses on binary association and composition links during the extraction of a document-oriented NoSQL database. However, it does not consider structured attributes, association classes as well as n-ary, generalization, and aggregation links.

To overcome the limits of these works, we have proposed a more complete solution which addresses different types of attributes and links: atomic and struc-



Fig. 8 Source model



Fig. 9 Target model

tured attributes, association (binary and n-ary), generalization, aggregation, and composition links as well as association classes.

5.4 Validation

Concerning the model extraction of schema-less NoSQL databases, our approach allows to display to the developer simultaneously a conceptual model and a physical model; the first to understand the semantics of the database and the second to write queries. To evaluate the relevance of our approach, our prototype (Sect. 4) was implemented by three developers at Trimane, a digital services company specialized in business intelligence and Big Data.

The three experienced developers (IT consulting engineers) were tasked with providing maintenance for three separate applications. None of the developers know, previously, the data model of the concerned applications. For each application, each developer writes 10 queries that have an increasing complexity according to three different cases: (1) without any data model, (2) with the physical data model, or (3) with the both conceptual and physical models. Figures 10a, b show respectively an example of the conceptual and physical models corresponding to one of the three applications. Note that due to lack of place, we present data models (conceptual and physical one) of only one application.

We should also highlight that for reasons of visibility, models are represented to the user in the same screen and with an appropriate format: JSON for the physical model and the graphic format for the conceptual one. Each time we click on a class on the conceptual model, we will have its equivalent on the physical model. For example, the part of the physical model written in bold corresponds to the selected class (Trials).

Each database is associated with a set of queries whose natural language statements are provided to the three developers. In Table 2, we calculated the average time of writing the queries by the three developers in each situation: (1) without any data model, (2) with the physical data model, or (3) with the both conceptual and physical models.

Our initial hypothesis was verified in the situations considered. This establishes that a knowledge of semantics and data structure allows the developer to write queries faster on a schema-less NoSQL database. The small difference noted between the use of the single physical diagram and the use of the two models (conceptual and physical) is probably due to the experience of the three developers.

6 Conclusion and Future Work

Our works are dealing with the reverse engineering mechanisms of schema-less NoSQL databases to provide users with models to manipulate this type of database.

Table 2 Query writing time

	Without model	Physical model alone	Conceptual and physical model
Developer 1	Database 1: 50 minutes	Database 2: 23 minutes	Database 3: 18 minutes
Developer 2	Database 2: 40 minutes	Database 3: 25 minutes	Database 1: 16 minutes
Developer 3	Database 3: 48 minutes	Database 1: 20 minutes	Database 2: 16 minutes
<i>Average</i>	<i>46 minutes</i>	<i>23 minutes</i>	<i>17 minutes</i>

We have presented in this paper an automatic process for mapping a UML conceptual model starting from a NoSQL physical model. We note that we have proposed, in previous works, a process to extract a NoSQL physical model starting from a document-oriented NoSQL database. So, we use this physical model to generate a conceptual model that makes it easier for developers and decision-makers to (1) understand how data are stored and related in the database and (2) write their queries. The mapping between the two models, physical and conceptual, is ensured by a set of transformation algorithms. Our solution addresses different types of attributes and links: atomic and structured attributes, association (binary and n-ary), generalization, aggregation and composition links, as well as association classes. We have experimented our process using a case study in the health care field. We have also validated our solution in three different real cases to prove that the generated conceptual model provides a good assistance to the user to express their queries on the database while saving a lot of time.

As future work, we plan to complete our transformation process to have more semantics in the conceptual model by considering other types of links such as reference links.

References

1. C.L.P. Chen, C.-Y. Zhang, Data-intensive applications, challenges, techniques and technologies: A survey on big data. *Inf. Sci.* **275**, 314–347 (2014)
2. A.B. Angadi, K.C. Gull, Growth of new Databases & Analysis of NOSQL Datastores. *International Journal of Advanced Research in Computer Science and Software Engineering* **3**, 1307–1319 (2013)
3. D.S. Ruiz, S.F. Morales, J.G. Molina, Inferring versioned schemas from NoSQL databases and its applications, in *International Conference on Conceptual Modeling*, (Springer, Cham, 2015)
4. C. Bondiombouy, Query processing in cloud multistore systems, in *BDA: Bases de Données Avancées*, (2015)
5. M.A. Baazizi, H.B. Lahmar, D. Colazzo, G. Ghelli, C. Sartiani, Schema inference for massive JSON datasets, in *Extending Database Technology (EDBT)*, 2017
6. M.A. Baazizi, D. Colazzo, G. Ghelli, C. Sartiani, Parametric schema inference for massive JSON datasets. *VLDB J.*, 1–25 (2019)
7. Extract Mongo Schema. <https://www.npmjs.com/package/extract-mongo-schema/v/0.2.9> Online; 5 October 2019
8. E. Gallinucci, M. Golfarelli, S. Rizzi, Schema profiling of document-oriented databases. *Inf. Syst.* **75**, 13–25 (2018)

9. M. Klettke, U. Störl, S. Scherzinger, Schema extraction and structural outlier detection for json-based nosql data stores. *Datenbanksysteme für Business, Technologie und Web (BTW 2015)* (2015)
10. B. Maity, A. Acharya, T. Goto, S. Sen, A framework to convert NoSQL to relational model, in *Proceedings of the 6th ACM/ACIS International Conference on Applied Computing and Information Technology*, (ACM, 2018), pp. 1–6
11. A. Brahim, R. Ferhat, G. Zurfluh, Model driven extraction of NoSQL databases schema: case of MongoDB, in *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 1: KDIR*, ISBN 978-989-758-382-7, pp. 145–154 (2019)
12. I. Comyn-Wattiau, J. Akoka, Model driven reverse engineering of NoSQL property graph databases: The case of Neo4j, in *2017 IEEE International Conference on Big Data (Big Data)*, (IEEE, 2017, December), pp. 453–458
13. J.L.C. Izquierdo, J. Cabot, JSONDiscoverer: Visualizing the schema lurking behind JSON documents. *Knowl.-Based Syst.* **103**, 52–55 (2016)
14. A.H. Chillón, D.S. Ruiz, J.G. Molina, S.F. Morales, A model-driven approach to generate schemas for object-document mappers. *IEEE Access* **7**, 59126–59142 (2019)
15. MongoDB, Mongoddb atlas database as a service. <https://www.mongodb.com/>. Online; 5 November 2019 (2018)
16. F. Budinsky, D. Steinberg, R. Ellersick, T.J. Grose, E. Merks, *Eclipse Modeling Framework: A developer's Guide* (Addison-Wesley Professional, 2004)

Composition of Parent–Child Cyberattack Models



Katia P. Maxwell, Mikel D. Petty, C. Daniel Colvett, Tymaine S. Whitaker, and Walter A. Cantrell

1 Introduction

Cybersecurity has become an urgent concern. Modern society is increasingly reliant on computer systems for nearly all aspects of life. There are many threats to those computer systems and the data they contain, including privacy invasion, financial theft, infrastructure sabotage, and election tampering. Motivated by the growing importance of cybersecurity issues, cybersecurity modeling is an active research area, with a wide range of applications and methods. Examples of cybersecurity modeling projects include cybersecurity risk using game theory, attacks on computer networks using distributed simulation, malware propagation through the use of agent-based modeling, evaluation of distributed denial-of-service detection processes using open-source network simulation software, and global positioning system (GPS)-related attacks on unnamed aerial vehicles (UAVs) using open-source UAV simulation test bed to name a few [1, 5, 8, 10, 18].

Hamadi and Benatallah proposed a Petri net-based algebra for modeling the composition of Petri nets representing Web services control flow [6]. They established a grammar and used Backus Normal Form (BNF) notation to describe the semantics of algebra operators used to combine the Petri nets. Web services was their target for Petri net designs, and the algebra grammar syntax was focused on its application

K. P. Maxwell (✉)
Athens State University, Athens, AL, USA
e-mail: katia.maxwell@athens.edu

M. D. Petty · C. D. Colvett · T. S. Whitaker
University of Alabama in Huntsville, Huntsville, AL, USA

W. A. Cantrell
Lipscomb University, Nashville, TN, USA

to the Web. Even though their research was geared toward Web services, the authors provided a general theoretical framework for Petri net composition.

Hamadi and Benatallah defined Petri nets composition operations based on Web service models containing a single start state and one final state [6]. The composition operations defined by Hamadi and Benatallah were: (1) sequential, (2) alternatively, (3) arbitrary sequence, (4) iteratively, (5) in parallel with communication taking place, (6) discriminatorily, (7) through dynamic selection, or (8) refinement.

A similar study in the area of Web Services and Petri nets was conducted by Yang et al. [23]. The authors use colored Petri nets to analyze the performance of behavioral properties for web services composition. Their focus is on the verification techniques which allow for designers to test and repair errors before running a service. The verification of their composition is done through traversing an XML document to show that the composition simulation is guaranteed to terminate, the composition which is based on model transformation yields a unique result, and it is found to be complete. The authors verify that the syntactic correctness can be presented by checking it against a generated colored Petri net metamodel. However, the authors do not have a unique way to verify the technique of semantic correctness or effectiveness of the transformation.

Most computer systems could be vulnerable to more than one type of attack, and so multiple component models must be combined to fully represent the system being attacked. Therefore, single (individual) cyberattacks that may occur need to be integrated or composed, to produce complete models of attacks to target computer systems. Doing so requires that the individual attack models be defined so as to be composable, that is when utilizing Petri net models, it must include places and transitions that serve as “connectors” and allow the models to be connected. The connectors must be designed to preserve the modeling semantics of the component models and pass needed stated information through the connections. To determine how to compose the cyberattack models based on the different attack models requires the understanding of the term “composability.” When it comes to research in the area of simulation, it has been documented that composability has different meanings dependent on the context in which it is noted [7, 11, 12, 20–22].

The definition of composability used in this study is a variation of the definition provided by Petty and Weisel [21]. Composability is the capability to select and assemble individual cyberattack models to form a complete cyberattack model on a target computer system based on specific requirements. In this paper, the process of taking two individual attack models and composing them is demonstrated. The unique characteristic of this process, however, is the relationship that is associated with the selected individual attack patterns. The patterns are documented as having a parent–child relationship. Following this introduction, Section 2 describes the Common Attack Pattern Enumerations and Classification (CAPEC) resource and the extensions of Petri nets used for cyberattack modeling. Section 3 discusses the composition methods for Petri nets with Players, Strategies, and Cost (PNPSC) previously published and describes the new method that focuses on the parent–child relationship between models. Section 4 provides an example of the composition between these types of models, and a final summary follows.

2 Background

The goal of this study is to expand in the methodologies of sequential and parallel composition described by Mayfield et al. [16]. The focus of this study is to determine how to compose cyberattacks aimed on specific target systems when individual attacks selected to be used in the composition have a parent–child relationship. To fully understand this study, there are two important concepts considered as a foundation before going into a component selection methodology. First is to understand the data that are used as the basis for designing components. In this study, the data are a report, named CAPEC report and described next, which is then converted to a PNPSC model.

2.1 CAPEC

There are several studies that focus on security threats. An instrumental concept for the design of secure software is the knowledge of cyberattacks [3]. Many attacks are documented as an execution flow report demonstrating a pattern that an attacker takes to perform their attack. These patterns can then be modeled to analyze the evolution of the attack and the impact that it will have. Some studies of modeling cybersecurity include projects in attacks on computer networks, malware propagation using agent-based modeling, and game theory [4, 9, 13]. In particular, MITRE Corporation developed an online database called the Common Attack Patterns Enumeration and Classification (CAPEC) which documents over 600 different attacks, how they are performed, their goals, and mitigations [17]. The CAPEC report does not specify if the attacks are performed by a single attacker or an organization, the assumption is made that the attacks can be completed by either. Even though there is no documentation on the number of attackers that are performing the attack, the patterns do describe different methods or techniques that can be used to achieve a specific attack goal. An attacker may prefer to run one attack over another due to their own knowledge of being able to control the process; however, since he or she has options, these must be considered. The attacker may combine two or more techniques or sections of them, creating a new attack pattern, to improve the chances of success, requiring a specific study of such situations.

The basic model under study, Petri nets with Players, Strategies, and Cost (PNPSC), is an enhanced view of the traditional Petri nets, including the analysis of actions taken by attackers and defenders, and the estimation of costs associated with those actions [15]. As stated by Petty et al., the formalism for PNPSCs has been determined and a method to automatically generate PNPSC cyberattack models based on available information from an attack in the CAPEC vulnerability database has been implemented and associated with a pattern completion score. The formalism and additional details of PNPSC nets can be found in the study cited in [15].

The goal of this study is to determine how to compose the cyberattack models based on the different components that are available with a parent–child relationship. When assembling the cyberattack for the target computer systems, the components will be selected from a repository. The repository will be accessed through a web application and based on user specifications, the components will be selected and assembled in the requested cyberattack model. These same components may also be reused in multiple cyberattack models for different target computer systems.

2.1.1 Levels of CAPEC Patterns

In order to demonstrate the composition, a review of characteristics of classification of CAPEC reports, original source of information for the composition process, is required. Each of the CAPEC attack patterns are classified into one of three levels: meta, standard, and detailed.

A *Meta level attack pattern* in CAPEC is a decidedly abstract characterization of a specific methodology or technique used in an attack. It is often void of a specific technology or implementation and is meant to provide an understanding of a high level approach. It can also be described as a generalization of related group of standard level attack patterns, being particularly useful for architecture and design level threat modeling exercises.

A *Standard level attack pattern* in CAPEC is focused on a specific methodology or technique used in an attack. It is often seen as a singular piece of a fully executed attack. A standard attack pattern is meant to provide sufficient details to understand the specific technique and how it attempts to accomplish a desired goal. It can be considered a specific type of a more abstract meta level attack pattern.

A *Detailed level attack pattern* in CAPEC provides a low level of detail, typically leveraging a specific technique targeting a specific technology, and expresses a complete execution flow. Detailed attack patterns are more specific than meta and standard attack patterns and often require a protection mechanism to mitigate actual attacks. A Detailed level attack pattern often will leverage a number of different standard level attack patterns chained together to accomplish a goal.

2.1.2 Inheritance: Parent–Child CAPEC Relationships

Many of the meta level attack patterns are a parent pattern to one or more child pattern(s). This relationship is documented under the section of the CAPEC attack pattern titled Relationship. An example of such information is shown in Fig. 1 for the attack pattern identified as CAPEC 26, Leveraging Race Conditions.

A similar view of this type of relationship is found in object-oriented programming described as inheritance. Inheritance is also associated with simulation

Nature	Type	ID	Name
ParentOf		27	<u>Leveraging Race Conditions via Symbolic Links</u>
ParentOf		29	<u>Leveraging Time-of-Check and Time-of-Use (TOCTOU) Race Conditions</u>

Fig. 1 Relationships of CAPEC 26 leveraging race conditions (version 3.1)

programming languages. Simula 67, which was developed in the 1960s, introduced the concept of inheritance [19]. As with real life, children will have characteristics of their parents and have some of their own that may not reflect the parent. The same idea is used in this study for the parent–child relationship in the CAPEC-reported attack patterns. There are two concepts of inheritance that can be used in this study: (1) single inheritance or (2) multilevel inheritance.

Single level inheritance is where subclasses (children) inherit the features of a superclass (parent). Therefore, a class inherits the properties of another class. Multilevel inheritance is where a subclass is inherited from another subclass [19]. While single inheritance is very straightforward in relation to a CAPEC-reported parent–child relationship, there are attack patterns that are children of one attack and are also a parent of another attack resulting in the multilevel inheritance.

2.2 *Petri nets with Players, Strategies, and Cost (PNPSC)*

In earlier work, an extension of Petri nets, referred as PNPS nets (Petri nets with Players and Strategies) was defined and applied to cyberattacks [24]. The PNPS formalism adds several features useful for modeling cyberattacks to Petri nets, including formalizations of the notions of competing players and their strategies. In this research program, the PNPS formalism was further extended in two ways: (1) a representation of the relative cost of the actions taken by the competing players was added and (2) some ambiguities in the original definition were resolved; the extended formalism is referred to as Petri nets with Players, Strategies, and Costs (PNPSC). The PNPSC formalism is able to model the essential elements of cyberattacks, including computer systems, their vulnerabilities, the actions taken by competing players to exploit or eliminate those vulnerabilities, and the relative costs of taking those actions [14, 15].

PNPSC nets model the dynamic states of system and the events that occur during an attack on that system as markings and transition firings in the PNPSC net respectively. A PNPSC net can have up to four different phases as part of its design, those phases are as follows: (1) Explore, (2) Experiment, (3) Exploit, and (4) Goals. The formalism models the attacker and defender as competing players who may independently observe the marking of a player-specific subset of the net, and based on the observed marking, act by changing the stochastic firing rates of a player-specific subset of the transitions, so as to achieve their competing goals. Transition

rate changes by a player incur a cost to that player. The PNPSC formalism includes several features of particular relevance to cyberattack modeling.

Firing Rates In a standard Petri net, the transition to fire is selected arbitrarily from among all enabled transitions. In a PNPSC net, each transition has an associated firing *rate*. The rate is interpreted as the number of times the transition will fire, on average, per time unit, or more generally, as the likelihood of the action or the event that the transition is modeling occurring. Higher rates result in increased likelihood of occurring. During each execution cycle, a firing time is generated for each enabled transition as an exponentially distributed random variate, using each transition's rate as the exponential distribution's rate parameter λ (see [2] for details). The enabled transition with the earliest firing time is selected for firing and simulation time is advanced to the current simulation time plus the selected transition's firing time.

Players, Player Goals, and Player-Observable Places Two (or more) competing (or cooperating) *players* are defined. The players have *goals*, defined as markings in the PNPSC net that they wish to achieve. They attempt to influence the sequence of firings, and thus ultimately the markings reached, in the PNPSC net in order to achieve their goals. Players do not have complete information during the execution of the PNPSC net. Each player may only observe a subset of the places during execution. Each player must determine what action(s) to take based on the current marking of their player-observable places. This feature models the limited information an attacker or defender might have regarding the state of the target computer system and the adversary's actions during a cyberattack.

Player-Controlled Transitions and Player Strategies Players attempt to influence the sequence of firings in the PNPSC net in order to reach a marking consistent with their goal. They do so by changing the firing rates associated with each transition. However, players may not change the rates of any transition in the PNPSC net. Rather, each player has a defined set of player-controlled transitions. A player may only change the rates associated with the transitions that player controls. The transitions controlled by a player represent those actions an attacker or defender may take or influence during a cyberattack. During execution, each player observes the PNPSC net's marking in that player's player-observable places, and based on the observed marking, may choose to change the rates of that player's player-controlled transitions. The mapping between the possible markings of a player's observable places and the changes to the player's controlled transition rates that the player will make in response to each marking is that player's *strategy*.

Costs of Player Actions Each action taken by a player has a cost that abstractly represents the time, effort, and skill level required, and expense incurred, to perform the action. Players incur costs in two ways. First, when a player changes the rate of a player-controlled transition, there is a cost proportional to the magnitude of the change. This is the cost of changing the probability of occurrence of the event represented by the transition. Second, when a transition actually fires, there may be

a cost to one or more players. This is the cost of the action or event that the transition represents.

Definition 1 A PNPSC net is formally defined as a 14-tuple $\text{PNPSC} = (P, T, W, M_0, B, L, G, \Theta, O, F, \Omega, \Gamma, C, D)$:

1. P, T, W, M_0, B, L as defined for a standard Petri net
2. $G = \{g_1, g_2, \dots\}$ finite, non-empty set of players
3. $\Theta = (T_0, T_1, T_2, \dots, T_{|G|})$; partition of transition set T into $|G| + 1$ subsets such that $\Theta = T_0 \cup T_1 \cup T_2 \cup \dots \cup T_{|G|}$ and $T_j \cap T_k = \emptyset$ for $0 \leq j, k \leq |G|$ and $j \neq k$; $T_i =$ set of transitions controlled by player g_i for $1 \leq i \leq |G|$ and $T_0 =$ set of stochastic transitions not controlled by any player
4. $O = (O_1, O_2, \dots, O_{|G|})$; collection of $|G|$ subsets of place set P , that is, $O_i \subseteq P$ for $1 \leq i \leq |G|$; O_i is the subset of place set P observable by player g_i
5. $F: T_0 \rightarrow \mathbb{R}^+$; fixed firing rates for non-player-controlled transitions
6. $\Omega: (T - T_0) \rightarrow (\mathbb{R}^+ \times \mathbb{R}^+)$; initial and maximum firing rates for player-controlled transitions
7. $\Gamma: (\Gamma_1, \Gamma_2, \dots, \Gamma_{|G|})$; collection of functions $\Gamma_i: M^*_{O_i} \rightarrow \mathbb{R}^{|T_i|}$ where Γ_i is a mapping from the possible markings of player g_i 's observable places to the desired firing rates for each of player g_i 's controlled transitions
8. $C = (C_{\text{fire}}, C_{\text{change}})$; where $C_{\text{fire}}: (T \rightarrow \mathbb{R}^+)$ is the cost for firing a transition and $C_{\text{change}}: (T \times \mathbb{R}^+) \rightarrow \mathbb{R}^+$ is the cost for changing the rate of a transition by $\delta \in \mathbb{R}^+$
9. $D: T \rightarrow \wp(G)$; players that incur a cost for a fired or changed transition

Θ is a partition of transition set T , which implies that no transition may be controlled by more than one player and some transitions may be controlled by no players. On the other hand, O is not necessarily a partition of place set P , thus places may be observed by 0, 1, or more than 1 player.

Γ represents the players' strategies. Given a marking M , each player g_i may observe the marking M_{O_i} of a subset O_i of the net's places. Based on that observed marking, player g_i will want to set the firing rates of the transitions T_i that g_i controls to certain values. Function Γ_i is thus a mapping from all possible markings of player g_i 's observable places, denoted $M^*_{O_i}$, to the desired rates for player g_i 's controlled transitions. Γ_i returns those rates as a vector with $|T_i|$ elements. C is the cost of the players' actions. C_{fire} is the cost of firing a transition; this represents the expense or effort required to perform the action the transition represents. C_{change} is the cost of changing the rate of a player-controlled transition. If a player changes the rates of multiple transitions, the rates are summed. Finally, D returns the set of players that incur the cost of firing a transition; that set may have 0, 1, or more than 1 player.

3 PNPSC Model Composition Methods

This study focuses on extending previous composition methods, sequential, parallel, and refinement, which are initially assumed as the most common ways of combining different cyberattacks and previously documented by Mayfield et al. [16]. The fourth method, which is the focus of this paper, is an extension of those methods to focus on a parent–child relationship between CAPEC attack patterns.

A sequential composition implies a possible failure of an attack technique (as a consequence of the attacker’s actions and not a defense mechanism) followed by a different approach. A parallel composition represents several attacks occurring concurrently, while refinement would allow the in-depth study of some model components by replacing them with its building blocks. The parent–child relationship composition is a little more involved than the sequential and parallel composition methods. The parent–child composition method is implemented mainly with attack patterns of a meta level, where other patterns are then able to be “plugged” into the meta level to eventually form a pattern that would reach the detailed level of a pattern.

To appropriately compose the parent–child PNPSC model the concept of “loop backs” are required for the inheritance concept of the design, these loop backs are added to the definition of an Extended Petri net (EPN). Based on Definition 1 the below definition provides the algorithm to implement the composition of PNPSC models with a parent–child relationship.

Definition 2 Given two PNPSC N_1 and N_2 defined by

1. $N_1 = (EPN_1, G_1, \Theta_1, O_1, F_1, \Omega_1, \Gamma_1, X_1)$ and $EPN_1 = (P_1, T_1, W_1, L_1, M_{01}, B_1, ES_1, EF_1)$
2. $N_2 = (EPN_2, G_2, \Theta_2, O_2, F_2, \Omega_2, \Gamma_2, X_2)$ and $EPN_2 = (P_2, T_2, W_2, L_2, M_{02}, B_2, ES_2, EF_2)$
3. The resulting PNPSC N of a parallel composition of N_1 and N_2 is defined by
4. $N = N_1 || N_2 = (EPN, G_I \cup G_2, \Theta_I \cup \Theta_2, O_I \cup O_2, F, \Omega_I \cup \Omega_2, \Gamma, X)$
5. With EPN defined as:
6. $EPN = (P, T, W, L, M_0, B, ES, EF)$ such that
7. $P = P_1 \cup P_2$
8. $T = T_1 \cup T_2$
9. $W = W_I \cup W_2 \cup \{(u,v) | v \in \{p | \nexists(x,p) \in (W_I \cup W_2), \forall x, \wedge v \in P_I \cap P_2\} \wedge u \in \{p | \exists(v,t) \wedge (t,u), (v,t) \in (W_I \cup W_2), (t,u) \in (W_I \cup W_2), t \in T\}\}$
10. $L = L_1 \cup L_2$
11. $M_0 = M_{01} \cup M_{02}$
12. $B = B_1 \cup B_2$
13. $ES = ES_1 \cup ES_2$
14. $EF = EF_1 \cup EF_2$
15. With F, Γ, X defined as:
16. $F: F_1 \cup F_2 \quad \Gamma: M(O_1 \cup O_2) \rightarrow \mathbb{R}$
17. $X: M(ES_1 \cup ES_2) \rightarrow \mathbb{R}$

4 Parent–Child Model Composition

For a parent–child relationship composition, inheritance is implemented in the composition of the attack pattern. This is done by incorporating the concept that all children will inherit the prerequisites of the parent in addition to their own. In the case that parent and child share techniques used in different phases or end goals, these will be combined into one in the resulting model diagram. This inheritance is used based on single models or multilevel.

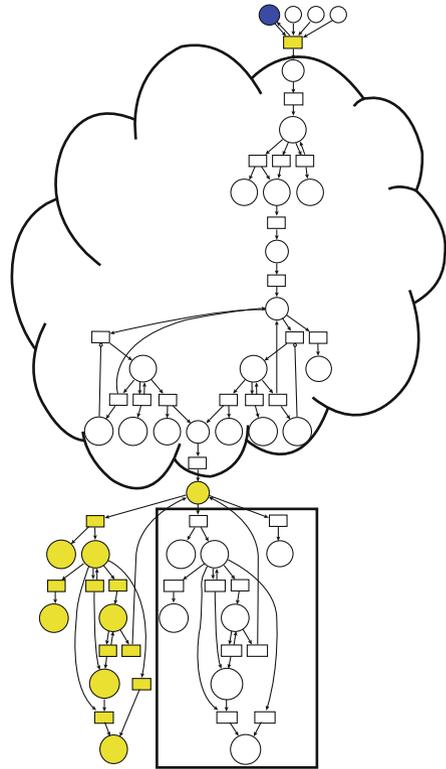
The design of these types of relationship occurs in three different ways. First, some of the parent attacks that are documented by CAPEC are of “meta” level as described above. In these scenarios, the phases of the children become the phases of the parent. Next, some of the other parents that are also marked as meta level have phase descriptions but not technique details. In these scenarios, the children still inherit the prerequisite knowledge requirement and are shown to run in parallel with the phases of the parent attack. Lastly, the parent is documented as a complete or detailed level pattern. This case works similar to the previous one where the children inherit prerequisite knowledge and are shown to run in parallel with the parent phases. If a child and parent or other children share techniques within phases, those techniques will be combined. This composition is displayed in the example below.

4.1 *Composition Example*

For this parent–child composition example, CAPEC 157 Sniffing has been selected to be represented as the parent. This particular attack pattern is the parent of multiple children, those child attacks are CAPEC 31 Accessing/Interception/Modifying HTTP Cookies, CAPEC 57 Utilizing REST’s Trust in the System Resource to Obtain Sensitive Data, CAPEC 65 Sniff Application Code, CAPEC 158 Sniffing Network Traffic, and CAPEC 609 Cellular Traffic Intercept. For the purposes of being able to graphically represent this composition, this example will focus on the composition of the parent attack with a single child, CAPEC 65 Sniff Application Code, as this is a complete detailed pattern and will allow for a complete representation of techniques within the attack phases.

Figure 2 (left) demonstrates a graphical view of the parent attack. Notice that this particular representation has a black cloud in the middle between the attack prerequisite knowledge requirement and the goals. The CAPEC attack pattern does not document techniques associated with explore, experiment, or exploit phases associated with this parent attack and therefore a black cloud of “uncertainty” is displayed. This black cloud of “uncertainty” will be replaced by the PNPSC model that represents the child attack pattern when composed with the parent pattern. By following the rules associated with inheritance, the child attack pattern will inherit the prerequisite of the parent along with possibly having its own set of prerequisite

Fig. 3 Parent–child composition (CAPEC 157 & 65)



the model, because this is a goal that can be achieved by the attacker having success in performing the Sniffing Application Code attack, this specific goal is not part of the ultimate goal of the parent attack, Sniffing, this additional goal is represented in the image with a square outline around it. The explore and experiment phases that are described by the child attack pattern have the cloud outline around it to represent that those are the sections that make up the design of the PNPSC model of the black cloud found in Fig. 2.

By creating the composition of the parent–child PNPSC models, a comprehensive cyberattack, already described in the CAPEC reports, can be represented. Instead of only focusing on a single attack, the composite model is representative of several methods in which a target system can be compromised, therefore allowing for developers, designers, systems administrators, and others to be able to better secure their systems.

5 Conclusion

Governments and different organizations have prioritized security against cyberattacks as a focus area for their operations. The simulation of cyberattacks allows for those who are responsible for securing the data and other aspects of their organizations to evaluate the defense mechanisms and the evolution of an attacker's strategy. Petri nets and different variations and extensions of Petri nets are being used to model cyberattacks described in CAPEC reports. This study briefly introduced the description of the CAPEC reports and the extension of Petri nets with Players, Strategies, and Cost. In this study, an additional composition method of PNPSC models was introduced to take into consideration attack patterns that have a parent-child relationship. The composition operation was formally defined in addition to previously defined methods of sequential and parallel compositions provided by Mayfield et al. An example showing the composition between Sniffing (parent attack) and Sniffing Application Code (child attack) was also discussed. With the use of these formal definitions, cyberattack models can be composed to represent an attack on a target system. These methods of composition will allow for the cyberattack models to be executed and simulated. The simulation of comprehensive cyberattacks on target computer systems will allow system personnel to make their systems more secure and plan mitigations to vulnerabilities in the case of an attack.

References

1. M. Ashtiani, M.A. Azgomi, A distributed simulation framework for modeling cyber attacks and the evaluation of security measures. *Simulation* **90**(9), 1071–1102 (2014)
2. J. Banks, J.S. Carson, B.L. Nelson, D.M. Nicol, *Discrete-Event System Simulation, Fifth Edition* (Prentice-Hall, Upper Saddle River, 2012)
3. S. Barnum, A. Sethi, Attack patterns as a knowledge resource for building secure software, in *OMG Software Assurance Workshop: Cigital*, (2007)
4. M.P. Fanti, M. Nolich, S. Simic, W. Ukovich, Modeling cyber attacks by stochastic games and timed Petri nets, in *IEEE International Conference on Systems, Man, and Cybernetics*, Budapest, 2016
5. T. Gamer, C.P. Mayer, Simulative evaluation of distributed attack detection in large-scale realistic environments. *Simulation* **87**(7), 630–647 (July 2011)
6. R. Hamadi, B. Benatallah, A Petri net-based model for web service composition, in *Proceedings of ADC'03*, volume 17 of CRPIT, Australia (2003)
7. S.M. Harkrider, W.H. Lunceford, Modeling and simulation composability, in *Proceedings of the 1999 Interservice/Industry Training, Simulation and Education Conference*, Orlando, 1999, pp. 876–881
8. S. Hosseini, M.A. Azgomi, A.R. Torkaman, Agent-based simulation of the dynamics of malware propagation in scale-free networks. *Simulation* **92**(7), 1071–1102 (2016)
9. B. Jasiul, M. Szpyrka, J. Sliwa, Detection and modeling of cyber attacks with Petri nets. *Entropy* **16**, 6602–6623 (2014)
10. A.Y. Javaid, F. Jahan, W. Sun, Analysis of Global Positioning System-based attacks and a novel Global Positioning System spoofing detection/mitigation algorithm for unmanned aerial vehicle simulation. *Simulation* **93**(5), 427–441 (2017)

11. JSIMS Composability Task Force (1997). JSIMS Composability Task Force Final Report
12. S. Kasputis, H.C. Ng, Composable simulations, in *Proceedings of the 2000 Winter Simulation Conference*, Orlando, 2000, pp. 1577–1584
13. I. Kottenko, E. Doynikova, The CAPEC based generator of attack scenarios for network security evaluation, in *The 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Warsaw, pp. 436–441, September 2015
14. K.P. Mayfield, M.D. Petty, Petri nets with players, strategies, and cost: A formalism for modeling cyberattacks, in *Proceedings of the 2018 International Conference on Security and Management*, Las Vegas, 30 July–2 August 2018
15. K.P. Mayfield, M.D. Petty, T.S. Whitaker, J.A. Bland, W.A. Cantrell, An extended Petri net formalism for modeling cyberattacks, in *Proceedings of the 2018 AlaSim International Conference and Exposition*, Huntsville, 22–23 May 2018, pp. 46–72
16. K.P. Mayfield, M.D. Petty, T.S. Whitaker, J.A. Bland, Composition of cyberattack models, in *Proceedings of the 31st International Conference on Computer Applications in Industry and Engineering (CAINE)*, New Orleans, 8–10 October 2018
17. MITRE. (n.d.). CAPEC - Common Attack Pattern Enumeration and Classification (CAPEC). Retrieved May 2017, from <https://capec.mitre.org/>
18. S. Musman, A. Turner, A game theoretic approach to cyber security risk management. *J. Def. Model. Simul. Appl. Methodology Technol.* (2017). <https://doi.org/10.1177/1548512917699724>
19. K. Nygaard, O.J. Dahl, The development of the simula languages, in *1978 ACM Special Issue: History of Programming Languages Conference*. SIGPLAN Notices Vol 13, No. 8, August 1978
20. E.H. Page, J.M. Opper, Theory and practice in user-composable simulation systems. Presentation for DARPA Advanced Simulation Technology Thrust, 1998
21. M.D. Petty, E.W. Weisel, A composability lexicon, in *Proceedings of the Spring 2003 Simulation Interoperability Workshop*, Orlando, 2003, pp. 181–187
22. D.R. Pratt, L.C. Ragusa, S. von der Lippe, Composability as an architecture driver, in *Proceedings of the 1999 Interservice/Industry Training, Simulation and Education Conference*, Orlando, 1999, pp. 882–891
23. Y. Yang, Q. Tan, Y. Xia, Verifying web services composition based on hierarchical colored Petri nets, in *Proceedings of the 1st International Workshop on Interoperability of Heterogeneous Information Systems*, Bremen, 2005
24. A.M. Zakrsewska, E.M. Ferragut, Modeling cyber conflicts using an extended Petri net formalism, in *Proceedings of the 2011 IEEE Symposium on Computational Intelligence in Cyber Security*, Paris, 11–15 April 2011, pp. 60–67, <https://doi.org/10.1109/CICYBS.2011.5949385>

Tree-Based Fixed Data Transmission for Healthcare Sensor Networks



Susumu Shibusawa and Toshiya Watanabe

1 Introduction

Lifestyle diseases are physical disorders that arise by continuing “unhealthy lifestyle habits.” To prevent lifestyle diseases, it is essential that a person adopt appropriate lifestyle habits in eating, exercise, rest, personal preferences, etc. Dementia, meanwhile, is a “state in which a person loses the ability to manage everyday living and social practices due to a chronic decline or vanishing of a variety of mental functions that developed normally after birth.” Here, diet and exercise are examples of behavior that have been shown to be somewhat effective in preventing dementia.

Modern medicine has come to rely for the most part on a reactive health system that provides treatment after the individual becomes sick. However, to overcome lifestyle diseases and chronic diseases and improve the quality of life, it is necessary to go beyond this reactive health model that responds after an illness begins [1]. The World Health Organization (WHO) defines health as “a state of complete physical, mental and social well-being and not merely the absence of disease or infirmity” [2]. Indeed, health is a dynamic state that is constantly changing based on one’s body, environment, and lifestyle and using technology to measure continuous change in an individual’s state of health can greatly improve human health. The time interval between an individual’s first visit to a medical clinic after the first signs of a disorder and the receiving of specialized treatment can be significantly shortened by using continuous multimodal sensors with good accuracy. Through statistical analysis,

S. Shibusawa (✉)

Ibaraki University, Hitachi, Ibaraki, Japan

e-mail: susumu.shibusawa.ptr@vc.ibaraki.ac.jp

T. Watanabe

National Institute of Technology, Gunma College, Maebashi, Gunma, Japan

e-mail: t.wat@ice.gunma-ct.ac.jp

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_42

593

daily activities collected by a sensor-network health support system can support early diagnosis, track the progress of chronic diseases, advice of various types of treatments, and recommend the adoption of healthy habits for many people [3, 4].

Data collected from sensors placed in the environment or attached to one's body generates time-series data that matches data with time points. Such time-series data obtained from physiological, environmental, and visual sensors can identify human behavior [5]. In medical treatment, sensor data related to a person's activities can provide valuable data for discerning the state of a patient with a chronic disease, the state of recovery of a patient undergoing rehabilitation, etc. [6, 7]. Wearable medical sensor systems have been applied to health monitoring, medical automation, human computer interaction, etc. [8]

Objects in the physical world need to be programmed using healthcare-oriented smart devices, network sensors, actuators, etc. A new distributed user interface for programming the dynamic relationships among smart objects has come to be researched for network-connected wearable devices and IoT devices [9]. Furthermore, to process IoT data from diverse types of computer environments in as short a time as possible, a calculation technique for performing data analysis near the source of that data must be developed [10].

Wide-ranged applications of wireless sensor networks create a need for efficient collection and processing of sensor data [11]. Data collection in a sensor network is an operation that collects data from all nodes in the starting node. Data aggregation, meanwhile, is an operation that calculates aggregated values of the data in all nodes [12]. Much research has been conducted on trees as the basis of data collection and data aggregation and their application [13]. Data aggregation in a sensor network with no fault aggregates data in the network toward a specific node along a spanning tree [14] configured within the network [15]. Most research of collective data processing including data collection and data aggregation in a wireless sensor network has come to use consumed energy as one evaluation criterion. Sensors close to the starting node must relay a large amount of data from sensors far from the starting node, so a large amount of energy is consumed. Up to now, several studies have been performed on a service to perform data aggregation in a distributed wireless environment [16, 17], the time complexity, message complexity, and energy cost of data collection and data aggregation in wireless sensor networks [12, 18], link scheduling [19], path planning [11], the throughput-delay trade-off [20], etc.

There is a need for effective studies on inter-node transmission and aggregation processing in a sensor network. In this paper, we introduce asynchronously operated fixed data transmission in a distributed environment and evaluate the execution times of fixed data transmission and level data transmission. Tree-based fixed data transmission can continue transmission operations with an average number of data at either a shallow-level edge or a deep-level edge as long as there is data to be transmitted. In contrast, level data transmission begins data transmission from leaves and performs data transmission in a more simultaneous manner from many edges, and then, at a shallow level near the root, transmits integrated data all at once. For a complete binary tree, the execution time of fixed data transmission with a maximum number of transmission data of 2 is equivalent to or smaller than the execution

time of level data transmission, and as the number of nodes increases, its speed approaches a value 1.5 times faster than level data transmission.

This paper is organized as follows. Section 2 provides preliminary information on data transmission in graphs. Section 3 introduces tree-based fixed data transmission and derives the execution time of data transmission. Section 4 provides the execution time of level data transmission and compares two data transmissions. Finally, Sect. 5 concludes the paper by describing the results of this study and touching upon future issues.

2 Preliminaries

A sensor network consisting of sensor nodes that generate data and nodes that send/receive and process data can be expressed as weighted graph $G = (V, E)$. Here, n node set $V = \{v_1, v_2, \dots, v_n\}$ represents the sensor nodes and nodes that send/receive and process data and m element set $E = \{e_1, e_2, \dots, e_m\}$ represents the set of edges between nodes. In addition, weight w of edge $e = (u, v) \in E$ for the two nodes $u, v \in V$ is a positive number and expresses distance between two points, unit-data transmission time, etc.

The data transmission time for sending data in node u to node v via edge $e = (u, v)$ in graph $G = (V, E)$ is defined as follows.

Definition 1 Given edge $e = (u, v)$ with weight w in graph $G = (V, E)$, time $t(p)$ of data transmission $u \rightarrow v$ for sending p data items in node u to node v is expressed as the product of edge weight and number of data items as follows.

$$t(p) = wp \tag{1}$$

□

Definition 2 (active nodes and active edges) Nodes and edges engaged in data transmission are called active nodes and active edges, respectively. □

Since a single node cannot engage in two data transmissions simultaneously, the two adjacent edges connected to that node cannot simultaneously engage in data transmission. In other words, active edges cannot be adjacent to each other.

Definition 3 (timeline diagram) A diagram that represents the sending/receiving of data between nodes in chronological order is called a “data-transmission timeline diagram.” □

A timeline has been used, for example, to express the order of public evacuation activities and the operations of related institutions at the time of a disaster in

chronological order [21]. The following presents an example of data transmission for the shortest paths [14] of a graph.

Example 1 Figure 1 shows an example of a weighted graph $G = (V, E)$. Here, alphabets represent nodes and figures represent edge weights. Figure 2 shows its shortest-path tree with s as starting node. For the case that each node of graph $G = (V, E)$ holds unit data, Fig. 3 shows an example of data transmission from each node to starting node s in the graph's shortest-path tree configured with s as root. In the figure, bold arrows show active edges in data transmission. Figure 4 shows the timeline diagram of this data transmission. In the figure, nodes are shown in the vertical direction and unit time intervals in the horizontal direction. Two filled circles connected by an arrow represent a pair of send/receive nodes and a pair of open and filled circles in the horizontal direction represents the range of time that two nodes connected by an arrow are engaged in sending and receiving. □

Fig. 1 Example of weighted graph $G = (V, E)$

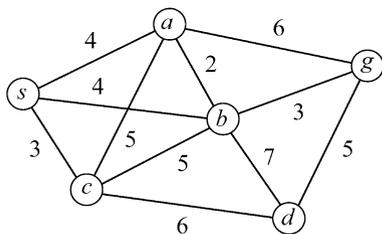


Fig. 2 Shortest-path tree of graph $G = (V, E)$

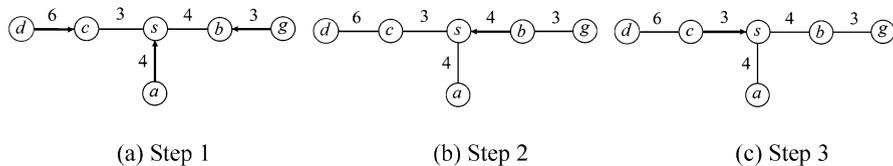
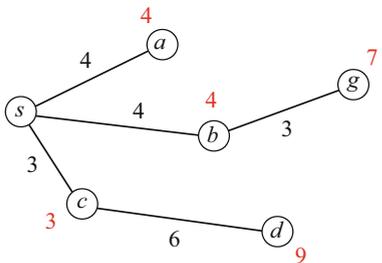


Fig. 3 Data transmission example in shortest-path tree of graph $G = (V, E)$. (a) Step 1. (b) Step 2. (c) Step 3

Fig. 4 Timeline diagram of data transmission example

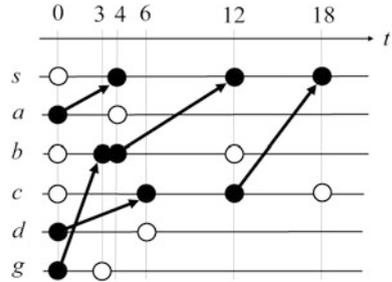
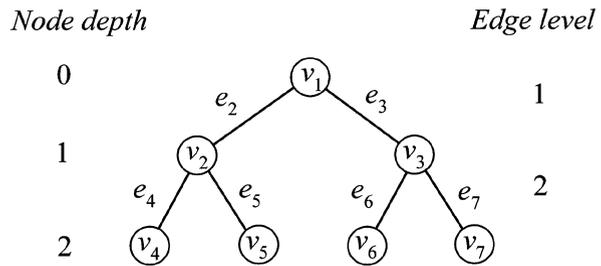


Fig. 5 Complete binary tree with $n = 2^3 - 1 = 7$ nodes



3 Tree-Based Fixed Data Transmission

Figure 5 shows a complete binary tree with $n = 2^3 - 1 = 7$ nodes. In this binary tree, nodes are numbered in the depth direction and from left to right at the same depth starting from root node v_1 . In addition, edge depth from the root is called a “level” and edges are numbered in level order and from left to right at the same level starting from e_2 .

Lemma 1 In a complete binary tree consisting of $n = 2^k - 1$ nodes (integer $k \geq 2$) such that edge weight is 1 and each node holds unit data, the data of each node is transmitted to root node v_1 . Denoting the truncation of a numerical value by the symbol $\lfloor \cdot \rfloor$, and treating the time taken for transmitting unit data from node v_i to parent node $v_{\lfloor i/2 \rfloor}$ via edge e_i with weight 1 as 1 unit time, the node with the largest data send/receive time T_{trans} is the root node with $T_{\text{trans}} = 2^k - 2 = n - 1$. The nodes with the next largest data send/receive time are child nodes v_2 and v_3 of root node v_1 with $T_{\text{trans}} = 2^k - 3 = n - 2$, and the data send/receive time of child nodes $v_4, v_5, v_6,$ and v_7 of nodes v_2 and v_3 is $T_{\text{trans}} = 2^{k-1} - 3 = (n - 5)/2$.

Proof The root node has $(n - 1)$ descendant nodes, and since each node holds unit data, the time taken for the root node to receive data from its child nodes is $T_{\text{trans}} = n - 1 = 2^k - 2$. The time taken for each of nodes v_2 and v_3 to receive data from its child nodes is $2^{k-1} - 2 = (n - 3)/2$ and the time taken for each of these nodes to send data to its parent node including its own data is $2^{k-1} - 1 = (n - 1)/2$, so total send/receive time of each of nodes v_2 and v_3 is $T_{\text{trans}} = 2^k - 3 = n - 2$. □

From Lemma 1, for the case that edge weights of a complete binary tree are equal and each node holds equal-size data, the node that incurs the largest data send/receive time is root node v_1 and the nodes that incur the next largest data send/receive time are the root's child nodes v_2 and v_3 . To therefore minimize data send/receive time, data transmission that prioritizes the root node is desirable. Even if transmission idle time were to be generated in nodes other than the root, transmission with no idle time generated in the root node is the most desirable. However, for a complete binary tree with equal edge weights, the difference in data send/receive time between root node v_1 and child nodes v_2 and v_3 is only 1 unit time and the data send/receive time of these three nodes is more than two times that of depth-2 nodes $v_4, v_5, v_6,$ and v_7 . For this reason, it is preferable to make the idle time of the three nodes $v_1, v_2,$ and v_3 small by adjusting the data transmission of the other nodes.

Given a complete binary tree with $n = 2^k - 1$ nodes (integer $k \geq 2$) and edge weight of 1, this section introduces a method for transmitting the unit data in all nodes to root node v_1 with a maximum number of transmission data of 2.

Example 2 Figure 6 shows an example of data transmission on a complete binary tree with $n = 2^3 - 1 = 7$ nodes in which the maximum number of transmission data is 2. In the figure, active edges that send data in each step are shown in bold and nodes that have completed transmission are marked with the "×" symbol. In addition, the number of transmission data along edge $e_i : v_i \rightarrow v_{\lfloor i/2 \rfloor}$ in step h is denoted as $f_h(i)$ and the number of data items held by receive node $v_{\lfloor i/2 \rfloor}$ after transmission is denoted as $g_h(\lfloor i/2 \rfloor)$. In a complete binary tree with $n = 7$ nodes, the number of data items held by a node other than the root node after data reception

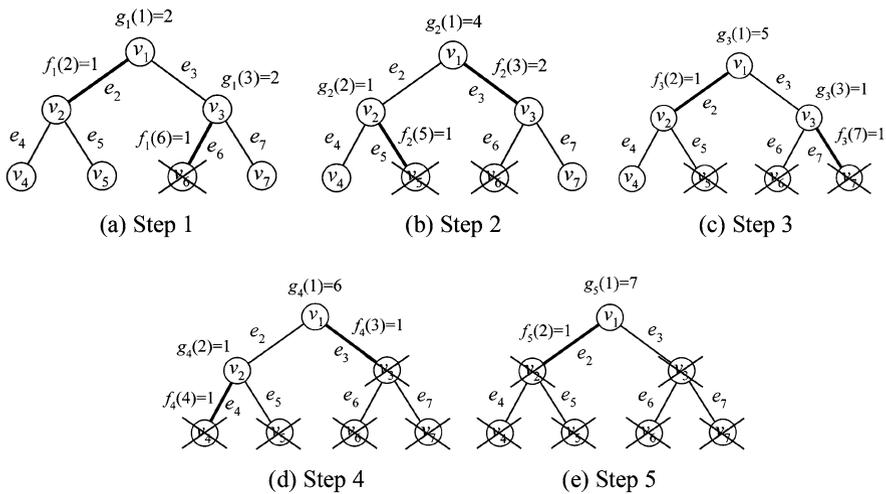


Fig. 6 Data transmission on a complete binary tree with $n = 2^3 - 1 = 7$ nodes and maximum number of transmission data of 2. (a) Step 1, (b) Step 2, (c) Step 3, (d) Step 4, (e) Step 5

is no more than 2, and the number of data items of all transmitting nodes after transmission is 0. After all data transmission is completed, the number of data items of the root node—including its own data—is equivalent to the number of nodes of the binary tree $n = 7$. Figure 7 shows the timeline diagram of the data transmission shown in Fig. 6. \square

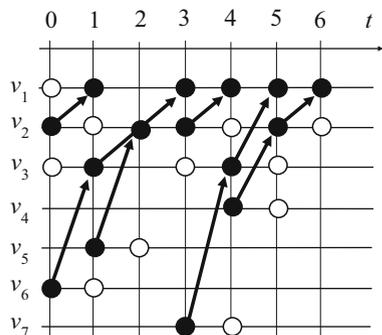
We call data transmission with a fixed number of data items transmitted via each edge at a time as simply “fixed data transmission.” The following outlines fixed data transmission operations for the case of a maximum number of transmission data of 2 in a complete binary tree $T = (V, E)$ having an edge weight of 1.

3.1 Outline of Fixed Data Transmission

1. The root node notifies each node of data transmission.
2. Root node v_1 performs data transmission alternately with each node via edges $e_2 = (v_1, v_2)$ and $e_3 = (v_1, v_3)$ connected to v_1 .
3. Node $v \in V$ cannot receive data from a child node while performing data transmission with its parent node.
4. While not performing data transmission with its parent, node $v \in V$ receives as many as two data items from its child node with the most data. If its child nodes have the same amount of data, data reception proceeds from the node whose last transmission is the oldest.
5. When the two child nodes of node $v \in V$ have completed transmission, v sends data to its parent node. Once node v has no more data, data transmission by v is completed. A leaf node completes data transmission after one transmission. \square

Lemma 2 Given that each node of a complete binary tree with $n = 2^k - 1$ nodes ($k > 2$) and an edge weight of 1 holds unit data, then, for data transmission in which the maximum number of transmission data is 2, the number of steps in which root node v_1 receives two data items is $(n - 5)/2$. In addition, the total number of steps in which the number of received data is 1 or 2 is $(n + 3)/2$. \square

Fig. 7 Timeline diagram of data transmission with maximum number of transmission data of 2 ($n = 7$)



Since adjacent edges do not engage in data sending/receiving simultaneously, level-1 and level-2 transmission edges must lie on different left/right subtrees so that root node v_1 is always receiving data. In the first four steps, level-1 and level-2 transmission edges can be combined in a total of four ways. That is, there are two ways in which edge e_2 can combine with an edge on level 2 of the right subtree and two ways in which edge e_3 can combine with an edge on level 2 of the left subtree.

Algorithm 1 Given that each node of a complete binary tree with $n = 2^k - 1$ nodes ($k > 2$) and an edge weight of 1 holds unit data, this algorithm performs data transmission with the maximum number of transmission data being 2 so that root node v_1 is always engaged in receiving data.

Input Each node holds unit data and the weights of all edges are equal.

Output Data in all nodes are sent to root node v_1 .

Method Outline of fixed data transmission. In the data transmission, steps are executed sequentially and levels can be executed concurrently. After repeating level-1 and level-2 transmissions $(n + 1)/2$ times, transmission $v_2 \rightarrow v_1$ is performed via edge e_2 . The final three steps consist of level-1 and level-2 transmissions only, so level 3 and deeper transmissions are repeated up to the $(n + 3)/2 - 3 = (n - 3)/2$ step. \square

Once data transmission begins under the fixed data transmission of Algorithm 1, node pairs that have completed preparations for transmission can perform asynchronous and distributed data transmission.

Theorem 1 Given that each node of a complete binary tree with $n = 2^k - 1$ nodes ($k \geq 2$) and an edge weight of 1 holds unit data, execution time $T(n)$ of Algorithm 1 in data transmission with a maximum number of transmission data of 2 and data-receive idle time $T_{idle}(n)$ of root node v_1 are as follows.

$$T(n) = \begin{cases} 2t(1) & (k = 2) \\ 4t(1) + \frac{n-5}{2}t(2) & (k \geq 3) \end{cases} \tag{2}$$

$$T_{idle}(n) = 0 \tag{3}$$

Here, symbols $t(1)$ and $t(2)$ denote the transmission time of 1 data item and 2 data items, respectively, between two nodes.

Proof For $k = 2$, the theorem holds since transmission of 1 data item occurs twice. There is no edge that can send or receive more data than edge e_2 or edge e_3 in each data transmission, so node v_1 never rests from data reception, which means that the receive idle time of v_1 is $T_{idle}(n) = 0$. Furthermore, for $k > 2$, from Lemma 2, node v_1 receives 1 data item four times and 2 data items $(n - 5)/2$ times, so the time $T_{trans}(n)$ that node v_1 is engaged in data reception is:

$$T_{\text{trans}}(n) = 4t(1) + \frac{n-5}{2}t(2) \tag{4}$$

The execution time of node v_1 in data reception is the sum of the time it is engaged in data reception and idle time as follows:

$$T(n) = T_{\text{trans}}(n) + T_{\text{idle}}(n) \tag{5}$$

□

Fixed data transmission can continue transmission operations with an average number of data items at either a shallow-level edge or deep-level edge as long as transmission data exist. In this transmission, for the root node that incurs the largest data send/receive time to have no idle time, two depth-1 nodes send data alternately to the root node, and four depth-2 nodes hold data transmitted from their descendant nodes and adjust the data transmission to depth-1 nodes.

4 Level Data Transmission

Next, we introduce level data transmission that transmits data at one-level intervals starting from edges on the bottom level connected to leaves on a binary tree. A level engaged in tree transmission is called an “active level.” The number of levels in a complete binary tree with $n = 2^k - 1$ nodes is $k - 1$, and in level data transmission, data transmission is performed in one-level intervals starting from level $k - 1$ connected to leaves. Here, a set of data transmissions via left/right edges is expressed as phase $j(1 \leq j \leq k - 1)$. Each phase consists of left-edge data transmissions and right-edge data transmissions, each of which is called a “step.” One phase consists of two steps and the step number, which increases sequentially from the start of data transmission, is denoted by the symbol $h(1 \leq h \leq 2(k - 1))$. Step h takes on an odd number or even number depending on whether the operation is left-edge or right-edge, respectively. This can be expressed as follows using phase parameter j :

$$h = \begin{cases} 2j - 1 & \text{(left edge)} \\ 2j & \text{(right edge)} \end{cases} \tag{6}$$

Example 3 Figure 8 shows an example of level data transmission on a complete binary tree with $n = 2^3 - 1 = 7$ nodes and edge weight of 1 in which each node holds unit data. In the figure, symbols $f_h(i)$ and $g_h(\lfloor i/2 \rfloor)$ denote the number of transmission data sent via edge $e_i : v_i \rightarrow v_{\lfloor i/2 \rfloor}$ in step h and the number of data held by receive node $v_{\lfloor i/2 \rfloor}$ after data transmission, respectively, and the “ \times ” symbol denotes nodes that have completed transmission. Here, if data transmission at one-level intervals is started from edges on the level connected to leaves, data transmission is performed via the left edges and right edges of level 2 in step 1 and

step 2, respectively, and via the left edge and right edge of level 1 in step 3 and step 4, respectively, thereby completing data transmission for this complete binary tree with $n = 2^3 - 1 = 7$ nodes in four steps. After step 2 is completed, nodes v_2 and v_3 each hold $(1 + 1) + 1 = 2 + 1 = 3$ items of data, and after step 4 is completed, node v_1 adds that data to its own data for a total of $(3 + 3) + 1 = 3 \times 2 + 1 = 7$ items of data. Figure 9 shows the timeline diagram of the data transmission shown in Fig. 8. □

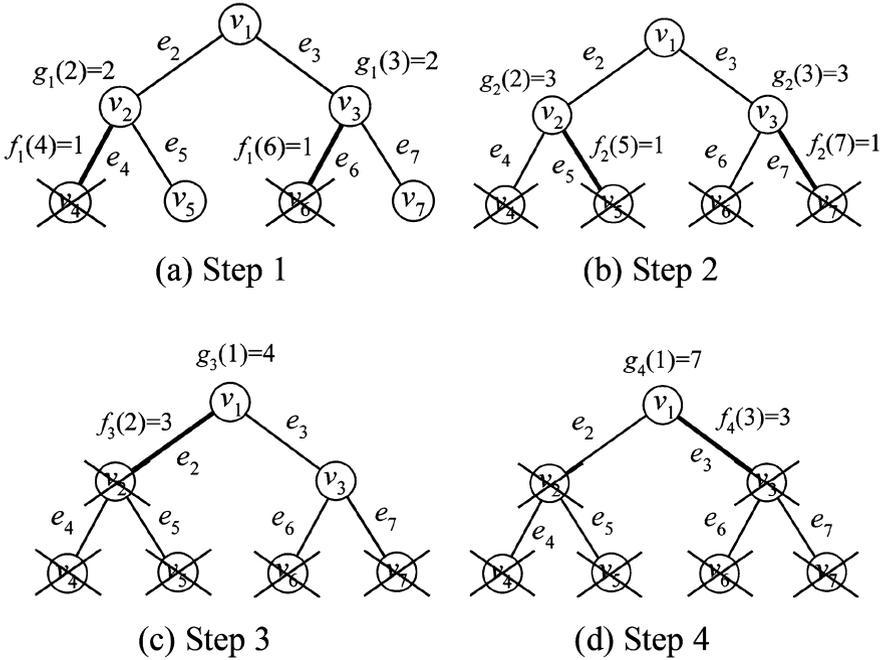
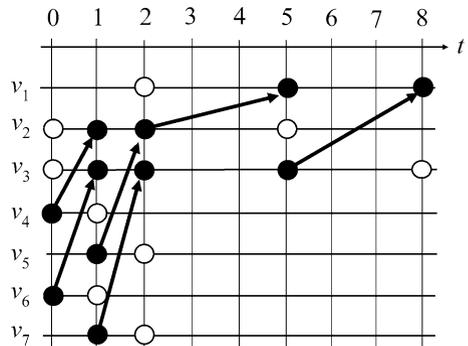


Fig. 8 Level data transmission on a complete binary tree with $n = 2^3 - 1 = 7$ nodes. (a) Step 1, (b) Step 2, (c) Step 3, (d) Step 4

Fig. 9 Timeline diagram of level data transmission ($n = 7$)



Property 1 Active level l consists of the following two cases for phase $j(1 \leq j \leq k - 1)$ in level data transmission on a complete binary tree with $n = 2^k - 1$ nodes:

1. $(k, j) = (\text{odd}, \text{odd}), (\text{even}, \text{even})$
Active levels consist of $(k-j)/2$ even levels, $l = k-j, k-j-2, \dots, 2$
2. $(k, j) = (\text{odd}, \text{even}), (\text{even}, \text{odd})$
Active levels consist of $(k-j + 1)/2$ odd levels, $l = k-j, k-j-2, \dots, 1$ □

Algorithm 2 Given that each node of a complete binary tree with $n = 2^k - 1$ nodes ($k > 2$) and an edge weight of 1 holds unit data, this algorithm sends the data of each node to the root node by level data transmission that performs data transmission at one-level intervals starting from level $k - 1$ connected to the leaves of the tree.

Input Each node holds unit data and the weights of all edges are equal.

Output Data in all nodes are sent to root node v_1 .

Method Procedure-level transmission of Fig. 10 that performs data transmission in one-level intervals starting from level $k - 1$ connected to the leaves of the tree. Parameter j and h denote phase and step, respectively. □

Once data transmission begins under the level data transmission of Algorithm 2, node pairs that have completed preparations for transmission can perform asynchronous and distributed data transmission.

Theorem 2 Given that each node of a complete binary tree with $n = 2^k - 1$ nodes ($k > 2$) and an edge weight of 1 holds unit data, execution time $T(n)$ of Algorithm 2 for level data transmission is as follows:

$$T(n) = 2 \left\{ t(1) + \sum_{i=0}^{k-3} t(3 \cdot 2^i) \right\} \tag{7}$$

Input: Each node holds unit data and the weights of all edges are equal.

Output: Data in all nodes are sent to root node v_1 .

Method: Procedure Level-transmission of Fig. 10 that performs data transmission in one-level intervals starting from level $k-1$ connected to the leaves of the tree. Parameter j and h denote phase and step, respectively.

procedure Level-transmission

Root node notifies each node of data transmission;

for $j=1$ to $k-1$ do

for $h=2j-1$ do

Send data from left children to their parent nodes via the left edges of the active level;

for $h=2j$ do

Send data from right children to their parent nodes via the right edges of the active level;

The parent nodes concatenate the data from their left and right children and prepare data of size $3 \cdot 2^{j-1}$;

Fig. 10 Procedure-level transmission

Here, the transmission time of p data is denoted as $t(p)$. For $k = 2$, only the first term on the right side of this equation holds.

Proof In the first phase of Algorithm 2, the number of transmission data of each left and right edge on that transmission level is 1, so data transmission time is $2t(1)$. Combined with the data originally held by the receive node, the number of data here becomes 3. In the second phase, the number of transmission data of each left and right edge on that transmission level is 3, so data transmission time is $2t(3)$ and the number of data of the receive node becomes $3 \cdot 2$. From here on, the number of transmission data continues to double every phase. In the $(k - 1)$ th phase, the number of transmission data of each left and right edge on level 1 is $3 \cdot 2^{k-3}$ and data transmission time is $2t(3 \cdot 2^{k-3})$. The execution time of data transmission can therefore be expressed as follows.

$$T(n) = 2 \left\{ t(1) + t(3) + t(3 \cdot 2) + \dots + t\left(3 \cdot 2^{k-3}\right) \right\}$$

□

The following relationship exists between the execution times of fixed data transmission and level data transmission for a complete binary tree.

Theorem 3 Given that each node of a complete binary tree with $n = 2^k - 1$ nodes and edges of constant weight w holds unit data, execution times $T^{\text{FIX}}(n)$ and $T^{\text{LEV}}(n)$ of fixed data transmission with a maximum number of transmission data of 2 and level data transmission, respectively, can be expressed as follows:

$$T^{\text{FIX}}(n) = (n - 1)w = (2^k - 2)w \tag{8}$$

$$T^{\text{LEV}}(n) = \frac{3n-5}{2}w = (3 \cdot 2^{k-1} - 4)w \tag{9}$$

Denoting the ratio of $T^{\text{LEV}}(n)$ to $T^{\text{FIX}}(n)$ as $T^{\text{L/F}}(n)$, $T^{\text{L/F}}(n)$ is expressed as follows:

$$T^{\text{L/F}}(n) = \frac{3n-5}{2(n-1)} = \frac{3 \cdot 2^{k-1} - 4}{2^k - 2} \tag{10}$$

The value of $T^{\text{L/F}}(n)$ approaches $3/2$ as the number of nodes n increases.

Proof From Definition 1, the time taken for sending p items of data via an edge of constant weight w is $t(p) = wp$, so from Theorem 1, we get:

$$T^{\text{FIX}}(n) = 4t(1) + \frac{n-5}{2}t(2) = (n-1)w$$

And from Theorem 2, we get:

$$T^{\text{LEV}}(n) = 2 \left(1 + 3 \sum_{i=0}^{k-3} 2^i \right) w = (3 \bullet 2^{k-1} - 4) w$$

□

In this section, we introduced level data transmission on a complete binary tree and compared the execution times of fixed data transmission and level data transmission. Although both transmission methods operate asynchronously in a distributed environment, data transmission in the case of level data transmission begins from leaves so that many edges perform data transmission simultaneously with the result that the number of data items transmitted at once becomes large at shallow levels near the root. In contrast, fixed data transmission can continue transmission operations with an average number of data items as long as transmission data exist.

Theorem 3 shows that the execution time ratio $T^{\text{L/F}}(n) \geq 1$ for all values of $n(n \geq 3)$ and the execution time of fixed data transmission with a maximum number of transmission data of 2 is equivalent to or smaller than that of level data transmission.

$$T^{\text{FIX}}(n) \leq T^{\text{LEV}}(n) \tag{11}$$

The value of execution time ratio $T^{\text{L/F}}(n)$ approaches $3/2$ as the number of nodes increases, and for a large number of nodes, fixed data transmission with a maximum number of transmission data of 2 is 1.5 times faster than level data transmission. This is because root-node idle time is 0 in fixed data transmission with a maximum number of transmission data of 2 but is not 0 in level data transmission. In future studies, there will be an even greater need to estimate the transmission loads applicable to both a centralized environment and distributed environment.

5 Conclusion

This paper evaluated the execution time of asynchronously operating fixed data transmission and level data transmission in a distributed environment. The execution time of fixed data transmission on a complete binary tree with a maximum number of transmission data of 2 is equivalent to or smaller than the execution time of level data transmission, and as the number of nodes increases, fixed data transmission with a maximum number of transmission data of 2 approaches a value 1.5 times faster than level data transmission. Going forward, consideration must be given to transmission method and data processing according to the usage environment of the sensor network.

This paper examined a binary tree having equal edge weights and an equal number of data held by each node, but attention must also be given to data transmission and data processing for various types of graphs. In addition, methods that introduce transmission delay on graph edges and express the relationship between

edge weight and transmission delay in an integrated manner is likewise another topic for future research. The development of an intuitive graphical representation of timeline diagrams is also desirable. An intuitive timeline diagram that makes it easier to grasp the number of transmission data and transmission paths would find application not only to analyzing edge and node usage (active edges and active nodes) in a sensor network but to other fields as well such as analyzing public evacuation routes and transport of goods at the time of a disaster [21]. Going forward, the research and technical development of real-time embedded algorithms for use in healthcare sensor systems is likewise desirable.

In the area of general sensor networks, there is a need for designing low-power wearable medical sensors and for researching and developing real-time on-sensor calculations for data aggregation. Furthermore, there is a need for advanced research of data collection latency in sensor networks and network-topology configuration and data routing.

Acknowledgments We would like to express our appreciation to Shota Suto of East Japan Institute of Technology Co., Ltd. for his helpful discussions over the course of this research. This work was partially supported by Grants-in-Aid for Scientific Research 17 K00746.

References

1. N. Nag, R. Jain, A navigational approach to health: Actionable guidance for improved quality of life. *IEEE Comput.* **52**(4), 12–20 (2019)
2. World Health Organization (WHO), Constitution, <https://www.who.int/about/who-we-are/constitution>. Accessed 8 June 2020
3. N. Zhu, T. Diethe, M. Camplani, et al., Bridging e-health and the internet of things: The SPHERE project. *IEEE Intell. Syst.* **30**(4), 39–46 (2015)
4. G. Sprint, D.J. Cook, R. Fritz, M. Schmitter-Edgecombe, Using smart homes to detect and analyze health events. *IEEE Comput.* **49**(11), 29–37 (2016)
5. T. Watanabe, K. Kamata, S.A. Hasan, et al., Design and implementation of an antagonistic exercise support system using a depth image sensor. *EAI Endorsed Trans. on Pervasive Health and Technology* **3**(10), c3 (2017)
6. F. Alvarez, M. Popak, V. Solachidis, et al., Behavior analysis through multimodal sensing for care of Parkinson's and Alzheimer's patients. *IEEE MultiMedia* **25**(1), 14–25 (2018)
7. D. Ravi, C. Wong, B. Lo, G.-Z. Yang, A deep learning approach to on-node sensor data analytics for mobile or wearable devices. *IEEE J. Biomed. Health Inform.* **21**(1), 56–64 (2017)
8. A. Mosenia, S. Sur-Kolay, A. Raghunathan, N.K. Jha, Wearable medical sensor-based system design: A survey. *IEEE Trans. Multi-Scale Computing Syst.* **3**(2), 124–138 (2017)
9. T. Kubitza, A. Schmidt, meSchup: A platform for programming interconnected smart things. *IEEE Comput.* **50**(11), 38–49 (2017)
10. R. Ranjan, O. Rana, S. Nepal, et al., The next grand challenges: Integrating the internet of things and data science. *IEEE Cloud Comput.* **5**(3), 12–26 (2018)
11. Y.-C. Wang, K.-C. Chen, Efficient path planning for a mobile sink to reliably gather data from sensors with diverse sensing rates and limited buffers. *IEEE Trans. Mob. Comput.* **18**(7), 1527–1540 (2019)
12. X.-Y. Li, Y. Wang, Y. Wang, Complexity of data collection, aggregation, and selection for wireless sensor networks. *IEEE Trans. Comput.* **60**(3), 386–399 (2011)

13. F.T. Leighton, *Introduction to Parallel Algorithms and Architectures – Arrays, Trees, Hypercubes* (Morgan Kaufmann, San Mateo, 1992)
14. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, *Introduction to Algorithms*, 3rd edn. (The MIT Press, Cambridge, MA, 2009)
15. S. Nath, P.B. Gibbons, S. Seshan, Z. Anderson, Synopsis diffusion for robust aggregation in sensor networks. *ACM Trans. Sens. Netw. (TOSN)* **4**(2), 7 (2008)
16. S. Madden, M.J. Franklin, J.M. Hellerstein, W. Hong, TAG: A Tiny AGgregation service for ad-hoc sensor networks, *ACM SIGOPS Operating Systems Review*, 36(SI) (2002)
17. S.R. Madden, M.J. Franklin, J.M. Hellerstein, W. Hong, TinyDB: An acquisitional query processing system for sensor networks. *ACM Trans. Database Syst. (TODS)* **30**(1), 122–173 (2005)
18. J. Crowcroft, M. Segal, L. Levin, Improved structures for data collection in static and mobile wireless sensor networks. *J. Heuristics* **21**(2), 233–256, Springer, New York (2015)
19. J. Ma, W. Lou, X.-Y. Li, Contiguous link scheduling for data aggregation in wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **25**(7), 1691–1701 (2014)
20. A. El Gamal, J. Mammen, B. Prabhakar, D. Shah, Throughput-delay trade-off in energy constrained wireless networks, *IEEE INFOCOM*, Hong Kong (2004)
21. Cabinet Office Japan, White Paper on Disaster Management 2019, http://www.bousai.go.jp/kaigirep/hakusho/pdf/R1_hakusho_english.pdf. Accessed 8 June 2020

Survey on Recent Active Learning Methods for Deep Learning



Azar Alizadeh, Pooya Tavallali, Mohammad R. Khosravi,
and Mukesh Singhal

1 Active Learning

Active learning is a subfield of machine learning that is also known as “query learning.” The idea in active learning is that it is possible to achieve higher accuracy if the active learning algorithm could select samples for its training procedure. The procedure of selecting samples can depend on the model. In most supervised learning models and their training algorithms, the assumption is that the dataset is labeled and to achieve a high accuracy a huge number of labeled data is available and computationally it is possible to train over the whole dataset. Sometimes such datasets and computational power are available, but that might not be the case in practical and real-world problems. In most real-world problems, the unlabeled data is abundant and labeling samples are expensive. Therefore, active learning is an interesting topic since in many real-world problems training over the whole dataset might not be possible or desirable for various reasons [35].

A. Alizadeh · P. Tavallali

Electrical Engineering and Computer Science, University of California, Merced, CA, USA
e-mail: aalizadeh@ucmerced.edu; ptavallali@ucmerced.edu

M. R. Khosravi

Department of Electrical and Electronic Engineering, Shiraz University of Technology, Shiraz,
Iran
e-mail: m.khosravi@sutech.ac.ir

M. Singhal (✉)

Electrical Engineering and Computer Science, University of California, Merced, CA, USA
Department of Electrical and Electronic Engineering, Shiraz University of Technology, Shiraz,
Iran
e-mail: msinghal@ucmerced.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_43

609

There are various scenarios in the literature of active learning. The main three base problems are explained here. The categorization here is based on a seminal survey [33].

1.1 Query Synthesis

In this setting the learner algorithm may ask for query of an unlabeled synthetic sample from the feature space [1]. This is among the very first methods of active learning. In this setting the learner will query a sample from the input space rather than querying an existing sample from dataset which is produced by some natural distribution of real samples. Such technique is understandable in low dimensions [2]. As a real-world example, the problem of predicting the absolute coordinates of a robot hand given the joint angles can be mentioned [8].

Conceptually speaking, query synthesis is a reasonable approach for understanding distribution of samples inside the input space. This is even tractable for specially low-dimensional spaces. For example, assume that there are two classes in a 2D space which is linearly separable. In such case the more samples produced around the separation boundary, the more informative queries will be. However, in high-dimensional problems, the results seem to be unrecognizable to human oracle, for example, classifying images of handwritten characters [3] through a neural network. However, such shortcomings can be tackled in some specific applications [21].

1.2 Stream-Based Active Learning

Stream-based or selective sampling [7] is when samples from dataset are observed sequentially and queried based on an informativeness measure. The assumption in this setting is that labeling a sample is free. The learner samples the data like a stream (one by one) and decides whether to query the sample for its label or not. The decision to query or not for an instance is cast through several ways. One strategy is to evaluate samples based on an informativeness measure or query strategy. Such decisions are biased random decisions such that they guarantee to query [9] more important samples. Another strategy is to explicitly compute a region of uncertainty [7, 29]. In practice for calculating such regions, several approximations are used [7, 10, 36].

1.3 Pool-Based Active Learning

In many datasets available, large number of samples are unlabeled data. The pool-based active learning [25] assumes that a small portion of the data is labeled while

a large part of it is not labeled. Query is done by selecting samples from the pool of data. The data is also assumed to be non-changing. Selection is done based on an informativeness measure to evaluate all the data. Afterward, the samples with highest measure of informativeness are selected. This can be based on the margin of samples from a decision boundary or cross-entropy.

The pool-based active learning is among the most popular methods of active learning and has a very old and rich literature and is studied for many applications such as video, image classification, and retrieval [16, 25, 28, 43].

Figure 1 represents schematic of mentioned active learning scenarios. In the rest of the paper, various query methods are represented, and then several deep learning with aspects of active learning are presented. Finally, the conclusion based on the explored papers are presented.

2 Query Methods

There are many query methods in the literature. In this section, several general approaches for query are presented.

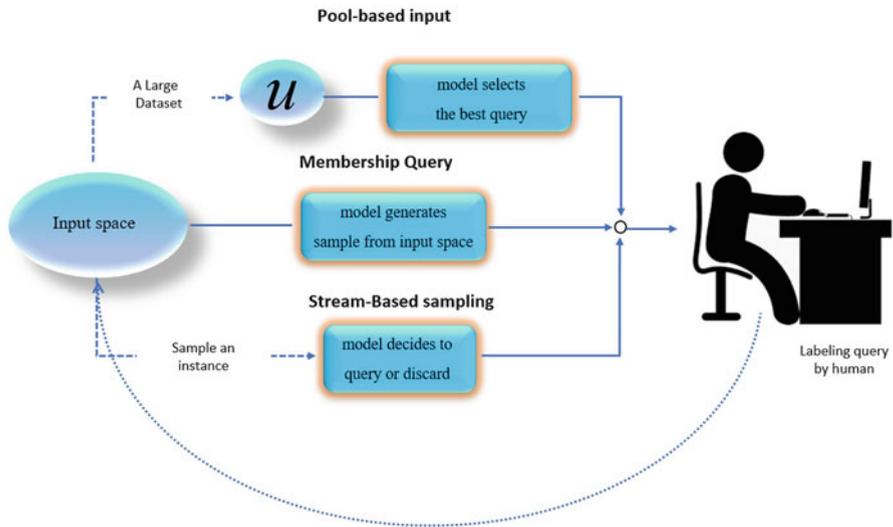


Fig. 1 Three base scenarios of active learning are presented

2.1 Uncertainty Sampling

Uncertainty sampling is among the most popular query frameworks [25]. Uncertainty sampling is based on querying the samples that the model has low certainty about their target label. Achieving the certainty measure is straightforward for probabilistic models. Such certainty measure can be of highest posterior probability [24, 25]. Another example is entropy function [37]

$$x^* = \operatorname{argmin}_x - \sum_k P(y_k|x; \theta) \log(P(y_k|x; \theta)) \quad (1)$$

where x , y_k , and θ are input samples, k th label, and parameters of the model. In fact objective function of (1) shows impurity of the prediction from a model for input of x . x^* represents the selected sample. In other words, the sample with highest impurity of outcome is selected for query. This measure can easily be used for various models and applications such as trees [17] and sequences [34]. Another example for certainty measure can be the instance whose best output target class is least confident:

$$x^* = \operatorname{argmin}_x P(y^*|x; \theta) \quad (2)$$

The objective function of problem (2) returns probability of most probable label. That is, the sample where y^* is

$$y^* = \max_k y_k. \quad (3)$$

2.2 Query-By-Committee (QBC)

Query-by-committee is based on gathering several methods that have different learned parameters over the labeled set. Each committee member is allowed to evaluate the dataset and produce its outcome. Here, the most informative samples are those that the committee members disagree the most on the outcome of the sample. Here, the challenge is how to gather various models that are different. Seung et al. [36] accomplishes this by sampling two random models that are consistent with labeled set. Another approach is to sample random models based on some posterior distribution $P(\theta|L)$ [28]. L is the labeled set. Mamitsuka et al. [27] proposed a method based on ensemble learning methods [12] and [6]:

$$x^* = \operatorname{argmax}_x - \sum_k \frac{V(y_k)}{C} \log\left(\frac{V(y_k)}{C}\right) \quad (4)$$

where $V(y_k)$ and C are the number of votes k^{th} label has received and number of committee members, respectively. As can be depicted, objective function of problem (4) has higher value for samples that raise high disagreement. Another disagreement measure based on Kullback-Leibler (KL) divergence [22] was proposed by [28]:

$$x^* = \operatorname{argmax}_x \frac{1}{C} \sum_{c=1}^C D(P_{\theta(c)} || P_C) \quad (5)$$

where $\theta(c)$ represents parameters of specific model number c . P_C is $P(y_k|x; C) = \frac{1}{C} \sum_{c=1}^C P(y_k|x; \theta(c))$. Here, $D(P_{\theta(c)} || P_C)$ is defined as follows:

$$D(P_{\theta(c)} || P_C) = \sum_k P(y_k|x; \theta(c)) \log\left(\frac{P(y_k|x; \theta(c))}{P(y_k|x; C)}\right) \quad (6)$$

2.3 Estimated Error Reduction

Various methods in the literature have also aimed at directly decreasing the generalization error. This is basically done by estimating the expected future error if a sample is queried. Therefore, the method tends to select samples that are more informative in the sense that they decrease estimated error. Roy and McCallum [31] proposed the first method for estimating expected error for text classification based on naive Bayes. Zhu et al. [47] combined this framework with a semi-supervised learning approach.

3 Recent Frameworks for Active Learning for Deep Learning

Recently, deep learning has become very useful in solving various machine learning tasks and problems. However, one major problem of training neural networks is existence of huge amount of labeled data. However, in most real-world problems, specifically in image processing and segmentation problems, there is huge amount of unlabeled data. Also, the best models of tackling such problems consist of training convolutional neural networks (CNNs). Therefore, the need for efficiently using dataset for training CNNs has become important. One solution to such problems is combining deep learning with concepts of active learning [42].

Beluch et al. [4] proposed a method based on uncertainty sampling. To do this, they propose a deep ensemble method based on [23, 30]. Their research has shown that ensemble methods perform better and lead to more calibrated predictive uncertainties that are latter evaluated in some acquisition functions. Experimentally,

they have shown their method performs better than Monte Carlo dropout [13] uncertainties and density-based approach [32].

Conventionally, acquisition functions are based on some criteria such as uncertainty sampling [13, 19, 45], space coverage through the data [32], and disagreement of a committee of models [18].

Sener and Savarese [32] proposes a density-based method to cover the entire space of input dataset. This is done using a geometric similarity function between images. The problem is defined as core-set selection. A set of points are chosen such that a model learned over the selected subset is competitive for the remaining dataset. They further characterize performance of any selected subset using the geometry of the data points.

Wang et al. [44] proposes using entropy of the softmax output for measuring the confidence of samples. Additionally, they label high confidence samples (pseudo-labeling). This method can be outperformed by [32].

Authors in [20] introduce a method based on estimating expected change of output of the neural network. However, this method is computationally expensive and performs similar to [44].

The method in [13] uses uncertainties for active learning algorithm over MNIST dataset and a medical dataset. The uncertainty estimates are based on sampling from the average of multiple softmax outputs of a sample through the network. Each time a random dropout mask known as Monte Carlo dropout is used.

In [46], authors propose a task-agnostic active learning method that works with the deep neural networks. The method is based on attaching a small parametric module to a target network. This module is called “loss prediction module.” The module learns to predict target losses of unlabeled inputs. Later, loss prediction module is used to suggest query of the data that the target model is likely to predict wrongly. This is a task-agnostic method since networks are learned from a single loss rather than of target values.

Authors of [45] proposed a semi-supervised batch mode multi-class algorithm for visual concept recognition. Their method is based on selecting data as diverse as possible based on a diversity constraint objective function, and then they propose optimization of this objective function through an efficient algorithm. Their problem is a discrete optimization problem. Similar approaches are proposed in [11, 14].

Methods in [5, 15, 26] are based on evaluating samples surrounding data points to choose one that can propagate the knowledge to the model.

In [38–41] authors use a margin-based algorithm for selecting samples from a pool of labeled samples. The assumption is that the samples are projected to binary space by the boosting formula. The margin exists in the binary space. The samples that are wrongly classified by the model are uniformly selected to be added to the training procedure.

Table 1 shows summary of several methods represented in this paper.

Table 1 Summary of deep learning methods presented here

Approach	Method	Discussion
Uncertainty sampling	[4]	The uncertainty is based on ensemble methods
	[13, 44–46]	Based on Monte Carlo dropout
Space coverage	[32]	A density-based method is used to cover the input space
Disagreement	[18]	The optimal training set is constructed by finding unlabeled scans which maximize the disagreement between our two complementary probabilistic models
Expected error	[20]	This method performs similar to [44] and is expensive
Selection casted as optimization	[11, 14, 45]	A discrete optimization problem solved through an efficient algorithm

4 Conclusion

Active learning is among the oldest and well-developed topics of machine learning. Recently, deep learning have shown breakthroughs in several topics of machine learning. However, the main drawback of deep learning techniques is the necessity for a huge labeled dataset. Most real-world datasets contain huge amount of unlabeled data, and labeling samples is very expensive. In many cases such as image processing problems, the well-known models for such problems are CNNs. Therefore, applying active learning techniques to CNN models for such tasks has recently gained attention. In this paper we first explained general active learning techniques, and then we explored recent advances in applying active learning methods to CNNs that tackle image processing tasks.

References

1. D. Angluin, Queries and concept learning. *Mach. Learn.* **2**(4), 319–342 (1988)
2. D. Angluin, Queries revisited, in *International Conference on Algorithmic Learning Theory* (Springer, 2001), pp. 12–31
3. E.B. Baum, K. Lang, Query learning can work poorly when a human Oracle is used, in *International Joint Conference on Neural Networks*, vol. 8 (1992), p. 8
4. W.H. Beluch, T. Genewein, A. Nürnberger, J.M. Köhler, The power of ensembles for active learning in image classification, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 9368–9377
5. M. Bilgic, L. Getoor, Link-based active learning, in *NIPS Workshop on Analyzing Networks and Learning with Graphs* (2009)
6. L. Breiman, Bagging predictors. *Mach. Learn.* **24**(2), 123–140 (1996)
7. D. Cohn, L. Atlas, R. Ladner, Improving generalization with active learning. *Mach. Learn.* **15**(2), 201–221 (1994)

8. D.A. Cohn, Z. Ghahramani, M.I. Jordan, Active learning with statistical models. *J. Artif. Intell. Res.* **4**, 129–145 (1996)
9. I. Dagan, S.P. Engelson, Committee-based sampling for training probabilistic classifiers, in *Machine Learning Proceedings 1995* (Elsevier, 1995), pp. 150–157
10. S. Dasgupta, D.J. Hsu, C. Monteleoni, A general agnostic active learning algorithm, in *Advances in Neural Information Processing Systems* (2008), pp. 353–360
11. E. Elhamifar, G. Sapiro, A. Yang, S. Shankar Sasrty, A convex optimization framework for active learning, in *Proceedings of the IEEE International Conference on Computer Vision* (2013), pp. 209–216
12. Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, in *European Conference on Computational Learning Theory* (Springer, 1995), pp. 23–37
13. Y. Gal, R. Islam, Z. Ghahramani, Deep Bayesian active learning with image data, in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70. JMLR.org (2017), pp. 1183–1192
14. Y. Guo, Active instance sampling via matrix partition, in *Advances in Neural Information Processing Systems* (2010), pp. 802–81
15. M. Hasan, A.K. Roy-Chowdhury, Context aware active learning of activity recognition models, in *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 4543–4551
16. S.C. Hoi, R. Jin, M.R. Lyu, Large-scale text categorization by batch mode active learning, in *Proceedings of the 15th International Conference on World Wide Web* (2006), pp. 633–642
17. R. Hwa, Sample selection for statistical parsing. *Comput. Linguist.* **30**(3), 253–276 (2004)
18. J.E. Iglesias, E. Konukoglu, A. Montillo, Z. Tu, A. Criminisi, Combining generative and discriminative models for semantic segmentation of ct scans via active learning, in *Biennial International Conference on Information Processing in Medical Imaging* (Springer, 2011), pp. 25–36
19. A.J. Joshi, F. Porikli, N. Papanikolopoulos, Multi-class active learning for image classification, in *2009 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2009), pp. 2372–2379
20. C. Käding, E. Rodner, A. Freytag, J. Denzler, Active and continuous exploration with deep neural networks and expected model output changes. arXiv preprint arXiv:1612.06129 (2016)
21. R.D. King, K.E. Whelan, F.M. Jones, P.G. Reiser, C.H. Bryant, S.H. Muggleton, D.B. Kell, S.G. Oliver, Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature* **427**(6971), 247–252 (2004)
22. S. Kullback, R.A. Leibler, On information and sufficiency. *Ann. Math. Stat.* **22**(1), 79–86 (1951)
23. B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, in *Advances in Neural Information Processing Systems* (2017), pp. 6402–6413
24. D.D. Lewis, J. Catlett, Heterogeneous uncertainty sampling for supervised learning, in *Machine Learning Proceedings 1994* (Elsevier, 1994), pp. 148–156
25. D.D. Lewis, W.A. Gale, A sequential algorithm for training text classifiers, in *SIGIR'94* (Springer, 1994), pp. 3–12
26. O. Mac Aodha, N.D. Campbell, J. Kautz, G.J. Brostow, Hierarchical subquery evaluation for active learning on a graph, in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014), pp. 564–571
27. N.A.H. Mamitsuka et al., Query learning strategies using boosting and bagging, in *Machine Learning: Proceedings of the Fifteenth International Conference (ICML'98)*, vol. 1 (Morgan Kaufmann Pub, 1998)
28. A.K. McCallum, K. Nigam, Employing em and pool-based active learning for text classification, in *Proceedings of International Conference on Machine Learning (ICML)* (Citeseer, 1998), pp. 359–367
29. T.M. Mitchell, Generalization as search. *Artif. Intell.* **18**(2), 203–226 (1982)

30. I. Osband, C. Blundell, A. Pritzel, B. Van Roy, Deep exploration via bootstrapped DQN, in *Advances in Neural Information Processing Systems* (2016), pp. 4026–4034
31. N. Roy, A. McCallum, Toward optimal active learning through sampling estimation of error reduction. *Int. Conf. Mach. Learn.*, Vol. 9; pp. 1–14 (2001)
32. O. Sener, S. Savarese, A geometric approach to active learning for convolutional neural networks. *arXiv preprint arXiv 1708.1* (2017)
33. B. Settles, Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences (2009)
34. B. Settles, M. Craven, An analysis of active learning strategies for sequence labeling tasks, in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* (2008), pp. 1070–1079
35. B. Settles, M. Craven, L. Friedland, Active learning with real annotation costs, in *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*, Vancouver (2008), pp. 1–10
36. H.S. Seung, M. Opper, H. Sompolinsky, Query by committee, in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* (1992), pp. 287–294
37. C.E. Shannon, A mathematical theory of communication. *Bell Syst. Tech. J.* **27**(3), 379–423 (1948)
38. P. Tavallali, M. Yazdi, Robust skin detector based on adaboost and statistical luminance features, in *2015 International Congress on Technology, Communication and Knowledge (ICTCK)* (IEEE, 2015), pp. 98–103
39. P. Tavallali, M. Yazdi, M.R. Khosravi, A systematic training procedure for viola-jones face detector in heterogeneous computing architecture. *Journal of grid computing.* **18**, 847–862 (2020). Springer Nature. <https://doi.org/10.1007/s10723-020-09517-z>
40. P. Tavallali, M. Yazdi, M.R. Khosravi, An efficient training procedure for viola-jones face detector, in *2017 International Conference on Computational Science and Computational Intelligence (CSCI)* (IEEE, 2017), pp. 828–831
41. P. Tavallali, M. Yazdi, M.R. Khosravi, Robust cascaded skin detector based on adaboost. *Multimedia Tools Appl.* **78**(2), 2599–2620 (2019)
42. S. Tong, active learning: Theory and applications, PhD dissertation. Stanford University/Press, California, USA, 2001 (August 2001). http://www.robotics.stanford.edu/~stong/papers/tong_thesis.pdf. Accessed 15 June 2021
43. S. Tong, D. Koller, Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.* **2**, 45–66 (2001)
44. K. Wang, D. Zhang, Y. Li, R. Zhang, L. Lin, Cost-effective active learning for deep image classification. *IEEE Trans. Circuits Syst. Video Technol.* **27**(12), 2591–2600 (2016)
45. Y. Yang, Z. Ma, F. Nie, X. Chang, A.G. Hauptmann, Multi-class active learning by uncertainty sampling with diversity maximization. *Int. J. Comput. Vis.* **113**(2), 113–127 (2015)
46. D. Yoo, I.S. Kweon, Learning loss for active learning, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 93–102
47. X. Zhu, J. Lafferty, Z. Ghahramani, Combining active learning and semi-supervised learning using gaussian fields and harmonic functions, in *ICML 2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, vol. 3 (2003)

Cloud-Edge Centric Service Provisioning in Smart City Using Internet of Things



Manoj Kumar Patra, Sampa Sahoo, Bibhudatta Sahoo,
and Ashok Kumar Turuk

1 Introduction

Cloud computing is nothing but the network of several servers and devices. In a cloud computing environment, multiple servers are connected over the Internet in a distributed manner which is responsible for providing computing resources. The service seekers request for computing resources to the server to execute their tasks. So, the cloud computing system consists of two parts: the front end composed of client devices such as mobile phones, tabs, and computers, and the backend composed of servers for computation and data storage. The frontend and backend are connected directly by wireless connectivity. Different types of services that a cloud computing technology can provide are Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). The way of handling various types of business needs by deploying ready-made software on the backend system in the cloud can be categorized as Software as a Service (SaaS). The deployment of the development environment on the cloud where one can write code, launch application, and test the applications is the concept of Platform as a Service (PaaS). The term edge computing refers to the distributed computing system that provides faster computation by bringing the computing device and storage closer to the client or where it required.

Internet of Things is a technology where different types of electronic devices are connected over the Internet and they communicate with each other. At the present time, smart cities are equipped with different types of electronic devices for smart mobility, smart parking, smart healthcare, smart security, smart governance, education, etc. Therefore, the smart city becomes smarter than before, and it requires

M. K. Patra (✉) · S. Sahoo · B. Sahoo · A. K. Turuk
Department of Computer Science & Engineering, National Institute of Technology, Rourkela,
Odisha, India

the real-time computation of requests coming from electronic devices in the smart city connected through the Internet of Things. For instance, suppose I am traveling from location X to Y in a smart city and I want to know the traffic status and shortest route that will take less time to reach the destination. My vehicle will communicate with the nearest data center with a request, and the server at the data center will consider the traffic status of every possible route along with the distance to the destination and provide me the best available route to the destination.

In the cloud-edge centric novel architecture model, service provisioning will be done both by edge device and using cloud servers. It integrates three technologies (cloud computing, edge computing, and the Internet of Things) together for better service provisioning in a smart city. Tasks are considered as a service request from the user and will be used interchangeably throughout this paper. The tasks are generated by IoT devices present in a smart city connected over the Internet. The tasks which require small-scale computing resources and memory will be dealt with the edge devices, and the large-size task requiring more computing resources and memory must be sent to the cloud server for computation. The real-time task is given priority over a normal task that does not affect much if the response comes with a little delay.

The main objective of this work is to propose a cloud-edge centric architecture that can be implemented for different service provisioning in a smart city using the Internet of Things. IaaS is used at cloud server and edge nodes. IaaS facilitates remote servers with computing capability, memory for storage, and networking. Edge devices will have comparably less computing power and storage space in it. A novel genetic algorithm (GA) has been proposed for service request assignment to different virtual machines in the edge and cloud layer. We have compared the system performance with and without edge layer architecture and found that the architecture with an edge layer between cloud and IoT layer performs better than without edge layer. The remaining section of the paper is organized as follows: after a brief introduction in Sect. 1, a few related works done so far which motivate us for this work is described in Sect. 2. In Sect. 3, the proposed architecture model and detailed description are presented with a different layer. Section 4 represents the experimental results, and finally, in Sect. 5, the conclusion is drawn with few future directions.

2 Background and Related Work

2.1 Smart City

There has been a huge demand in the last few years for the smart city which is sustainable, smart, effective, efficient, interconnected, self-repairing, robust, and adaptive. The term “smart city” has been used by many researchers, academician, and even a number of companies that refers to the integration and provisioning

of several smart services to the citizens in an urban ecosystem. A smart city is a strategic approach to integrating technologies for sustainability. A smart city uses information and communications technology (ICTs) to digitally connect, optimize, and deliver the smart services to the citizens and hence improve the quality of life. The main objective of developing a smart city is to provide smart services to citizens. Now, the question arises: What are smart services? The term smart service refers to incorporating innovative technologies, both in terms of software and hardware. Some examples of smart services are smart billing systems, smart meter, real-time monitoring of water quality online, a network of video surveillance around the city, smart parking, etc.

Now, another question arises: What is the requirement of a smart city and how important it is? It is predicted that by 2030, 60% of the population of the world will be living in cities. That forces us to think about the heavy shortage of energy, huge traffic that will lead to transportation problems, water management issues, construction of roads and buildings, and public space management, and many more issues will be raised. At the same time, we will have to ensure the overall sustainability of the environment, society, and economy. Broad cooperation is needed for smart city development; a partnership among citizens, collaboration among organizations, teamwork, integration of modern technologies are few things that can make a city smarter and make living easier for citizens.

There has been a tremendous effort in recent years for the development of smart cities all over the world to provide efficient public services, better mobility, resource management, security, clean air and water, and overall a better urban ecosystem. The main features of a smart city are

- Collecting information about the events taking place around the city.
- Efficient traffic management, i.e., availability of streamlined transportation system with better connectivity and optimized traffic.
- Providing variety of transportation options.
- Providing sustainable environment and waste management.
- Efficient allocation of land for construction of road, schools, building, open spaces, playground, etc.
- Integrated ICT, security, and surveillance.
- Transparency and open access to data.
- Availability of smart cooling and heating system in intelligent building saving water and energy usage and reducing maintenance cost of the building.
- Real-time availability of information from different locations in a city.

To support the above features of smart city, we can make use of different advanced technologies developed in recent years, and several smart devices will be an added advantage.

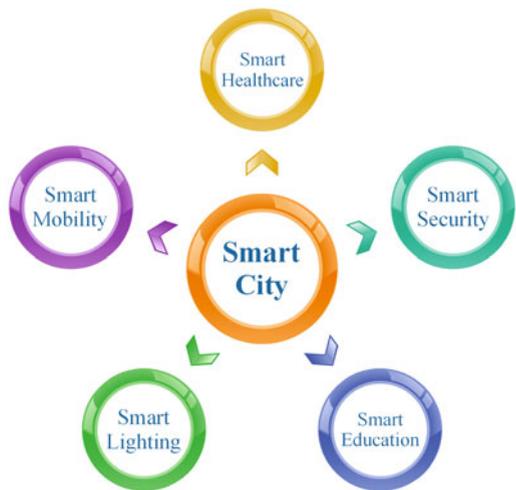
The collection and provisioning of real-time information from different parts of a city plays a vital role in smart service provisioning and using real-time applications. In a smart healthcare system, a smart service can be monitoring the physical activity of a patient in a hospital. The smart device present in the room will study the gesture of the patient and adjust the smart bed to make him comfortable. An

automatic alarm system is an example of smart security. Technologies such as image processing and video processing can be used for smart security in a smart city. Smart technology is the backbone of smart education. The use and implementation of smart devices such as laptops, smartphones, iPods, etc. in the classroom will help the students in understanding the subject and make them interesting. In a smart lighting system, based on daylight availability, the electric lighting will be adjusted to reduce energy consumption. Different light sensors such as photovoltaic light sensor, light-dependent sensor, and proximity light sensor can be used to detect the intensity of light. In a smart mobility system, real-time travel information such as availability of buses with arrival and departure time will be provided to the citizens so that they can plan their trip on public transport. Another example of smart mobility is finding available parking space while traveling. The system will analyze the real-time data and promptly inform the driver about where the free space is available. A brief overview of different smart services in a smart city is presented in Fig. 1.

2.2 Related Work

A lot of research has been going on in the field of cloud computing, edge computing, and the Internet of Things. The efficient use of these three technologies in a smart city for better service provisioning is a key research challenge and one of the most talked trends in the recent past. Swati Dhingra et al. in [1] proposed an approach for smart traffic monitoring while considering latency and response time. The smart traffic monitoring system takes care of traffic light and also the congestion level. The performance is analyzed by integrating cloud, fog, and Twitter and observed the improved performance. Tweet message is being used for the real-time alert

Fig. 1 Smart services in a smart city



about congestion on the road. Wei-Hsun Lee et al. in [2] designed a system that smartly manages the traffic signal. That is useful for intelligently controlling the transportation system in a smart city. The most important aspect of the proposed system is the roadside unit controller. The system is designed in such a way that it gives priority to the vehicle with public service and it also gives information about in which direction the emergency vehicles are moving. Daming Li et al. in [3] proposed a genetic algorithm-based vehicle management system that uses image perception as input and tried to improve the smart management and operational efficiency of the smart city.

Shuo Tian et al. in [4] listed out various technologies that can be used for smart healthcare in a smart city. They also presented the current status of smart healthcare, i.e., what is the technology being used in different areas such as risk motoring, virtual assistants, smart hospitals, drug research, etc. Some of the key challenges such as macro-guidance and programmatic documents that create an uncertain goal of development and finally cause resource wastage are pointed out. Akshi Kumar in [5] proposed a method to detect healthcare-related duplicate message pairs to overcome the matching problem of a semantic question in the smart healthcare system. A hybrid deep learning model has been used that uses a multi-layer perceptron and Bi-LSTM Siamese neural network to find the similarity between two messages. After finding the probability of similarity, Euclidean distance is used to find the similarity between two messages. A framework using cloud computing for health monitoring has been proposed by Abdulhamit Subasi et al. in [6].

Janakiramaiah B et al. in [7] proposed a convolution neural network-based intelligent video surveillance system for smart security in a smart city. The main purpose of the system is to generate an alert in case of any emergency or when an abnormal event occurs. The automatic task analysis from a video and finding the normal and abnormal sequence of images are being done, and if the analyzed activity is abnormal, an alert will be generated. The proposed system achieved a very low false alarm rate with trained CNN. G. Baldoni et al. in [8] presented a capillary platform for smart video surveillance that supports plug-and-play and is dynamic, and the number of devices is flexible as well as scalable. It's a distributed architecture with several network access nodes which connect multiple smart access nodes. The main advantages of this platform are network traffic reduction resulting in improved performance, low latency, and platform add-ons. A machine learning-based smart surveillance system using a distributed system with edge node closer to the IoT devices has been proposed by Paolo Bellavista et al. in [9], where each edge node learns the shared model and keeps track of the local knowledge. The performance of the proposed system is analyzed using advanced machine learning algorithms and found to be good.

Abdulhameed Alelaiwi et al. in [10] focused on how co-learning in engineering education can be facilitated using smart devices. The proposed work can be used in a smart city environment for providing smart education to the students. The authors discussed some smart technologies which can be used for a smart classroom such as multi-screen display, real-time mixer, and secure access to contents. In [11], authors focused on the interoperability of smart devices in the smart classroom for

seamless communications. For a particular domain, the concept is structured using ontology to solve the problem of the simultaneous operation of devices. It uses learner analytic and context information to emphasize the progress of a student. Authors in [12] tried to exploit cloud and edge computing for the Internet of Things by an osmotic architecture that uses deep learning for smart classroom.

Even though a lot of work has been done in the field of abovementioned three technologies, still there is the scope of improvement due to the lack of a standard for IoT environment and ever-developing cloud and edge computing. The integration of IoT with cloud and edge computing significantly affects the traditional view of cloud computing. The increased number of IoT devices will generate a huge amount of data in different formats such as discrete, continuous, binary, image, audio, video, etc. The IoT devices will be sending those data to the cloud continuously, so how to handle this Big Data in the cloud is another research challenge.

3 Cloud-Edge Centric IoT Architecture

This section describes the proposed cloud-edge centric architecture for the smart city using the Internet of Things. The architecture consists of three layers, namely, the IoT layer, edge layer, and cloud layer. The bottom-most layer is the IoT layer where the Internet of Things is used to connect smart services, on top of it edge layer is presented, and the cloud layer is the topmost layer. The detailed architecture model is described diagrammatically in Fig. 2.

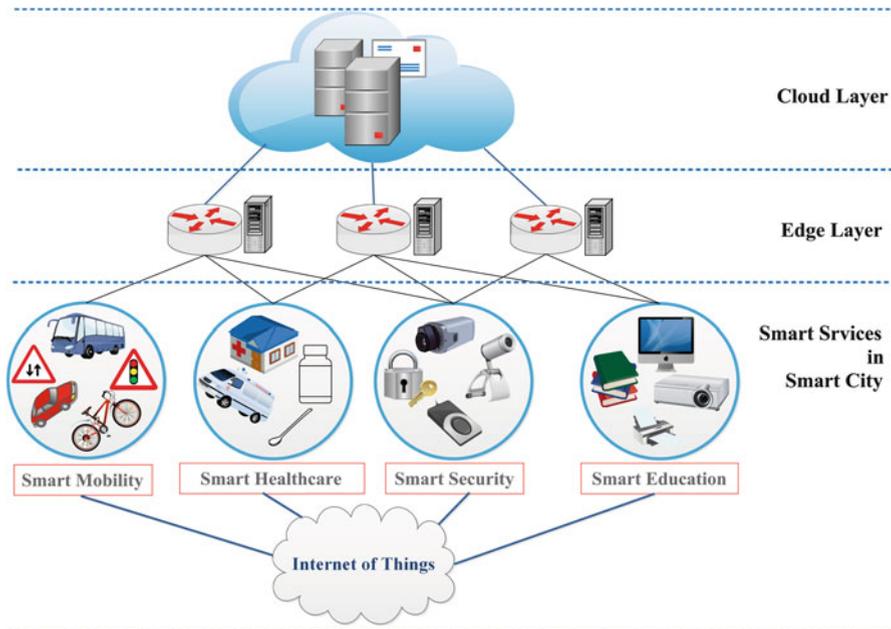


Fig. 2 Proposed cloud-edge centric architecture

3.1 Cloud Layer

The cloud layer is designed to establish communication among IoT devices and to process the users' requests. In the cloud layer, multiple servers are deployed and connected. Servers in the cloud layer are having large storage devices and better computing capability and offer on-demand remote virtual processing environments. However, since the cloud servers are remotely placed from the IoT devices, latency will increase with an increase in distance. Users' private data is transferred to the server in the cloud layer through the Internet, so there is a possibility of data loss or cyberattacks. Interruption in network connection is another issue that needs to be addressed to make sure that there will not be any problem in the network as data transfer takes place over the Internet. Cloud layer is designed for processing users' request that requires more memory space and computational resource.

3.2 Edge Layer

The edge layer is designed to bring computing capability near the service seekers where each edge node is having virtual machine (VM) with less storage and computing capability. If the service request can be completed in the edge layer, why go for the cloud layer? Adding an edge layer to the architecture may be a little more expensive and it's not that scalable as cloud, but the edge device near to the user will provide immediate response reducing latency, so there will not be any problem with bandwidth and very less chance of having trouble with the loss of connection. Since the request will be processed by the edge node near to the user, no need to transfer data over the Internet. Hence, data will be highly secured and user experience will be improved. Edge node also consumes less power. So, the edge layer is designed for processing users' request that requires less memory space and computational resource and high latency requirement.

3.3 IoT Layer

The IoT layer is the bottommost layer, and it is responsible for connecting all smart devices and smart services in a smart city. Different smart services such as smart mobility, smart healthcare, smart security, smart education, etc. consist of a lot of smart devices that are connected over the Internet and using the Internet of Things technology. In a smart mobility or transportation system, the smart device can be a car, bike, sensors, camera devices, etc. Few challenges in smart transportation systems are automatic vehicle notification, automatic road enforcement, automatic vehicle speed detection to make sure that a vehicle is moving in the legal speed limit, vehicle re-identification, collision avoidance system, etc.

In a smart healthcare system, different smart services that can be provided are continuous glucose monitoring that will help the diabetics to automatically monitor the blood glucose level. The ingestible sensors can be used to improve the patients' medication by tracking them regularly. Smart sensors can be used to monitor the room temperature of a patient and adjust them according to the patient's comfortability. Smart cameras are connected over IoT to monitor patients' movement on the bed and give an alert when a patient is uncomfortable on the bed. All the smart devices will send real-time data to the central data processing unit [14]. Whenever any service request comes from a doctor about the patient, the data processing unit will process the data to provide real-time information about the patient.

So, the IoT layer is designed for collecting data from different locations using several smart devices. A huge amount of sensor data will be collected by IoT devices and transferred to the cloud continuously.

3.4 VM Model

A set of virtual machines $V = \{v_1, v_2, \dots, v_m\}$ are running in the cloud layer as well as edge layer in the proposed architecture. Each virtual machine v_j has its own computing capability or speed and is represented by sp_j , $j = 1, 2, \dots, m$. The VM in the cloud layer is having more storage and computing capability than a VM in edge layer. The VM in the cloud layer is designed for execution of large tasks, and the VM in the edge layer is designed for executing small size tasks to provide an immediate response to the user with low latency.

3.5 Task Model

The detailed description of the task is presented by task model. Let the set $T = \{t_1, t_2, \dots, t_n\}$ represent the set of independent tasks generated using Poisson distribution [13]. Each task has the following attributes: $t_i = \{at_i, s_i, d_i\}$, where at_i is the arrival time, s_i is the size in MI (million instruction), and d_i is the deadline of the i th task. The expected execution (ET) time of task t_i can be calculated as

$$ET_i = \frac{s_i}{sp_j} \quad (1)$$

where s_i is the size of the i th task and sp_j is the speed of j th VM.

The completion time (CT) of a task can be calculated as

$$CT_i = WT_i + ET_i \quad (2)$$

where, WT_i is the waiting time of task t_i .

4 Experimental Analysis and Results

As stated earlier there are two types of tasks: one that will be executed at the edge which is small in size and requires quick response and another one that will be executed in a cloud server which is large in size. Two different queues are maintained to schedule the tasks. The tasks are assigned to different virtual machines using a novel genetic algorithm.

4.1 Proposed Genetic Algorithm for Task Assignment

The developed genetic algorithm not only assigns the task to a particular VM but also considers the deadline of the task. Each VM is considered as a gene and a set of 10 VM is considered as a chromosome. The work-flow of the proposed algorithm is given in Algorithm 1.

The structure of a chromosome and how the tasks are being assigned to the different virtual machines in a chromosome are shown in Fig. 3. Each element (gene) in the chromosome represents a virtual machine. The tasks are being mapped to a virtual machine based on the estimated completion time and the fitness value.

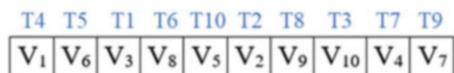
Algorithm 1 Genetic Algorithm Work-flow

- 1: Select the initial population, i.e., fix the number of chromosomes m and number iterations p .
 - 2: Repeat steps 3–6 for p times.
 - 3: Calculate the fitness value of each chromosome.
 - 4: Select m chromosome based on the fitness value.
 - 5: Perform crossover among the selected chromosomes to produce new chromosome.
 - 6: Perform mutation on newly produced chromosome.
 - 7: Return the best chromosome.
-

Initial Population Size and Number of Iteration The experiment was performed by taking the initial population size 30, 40, 50, and 60. The best result was observed with the initial population size 50. So the population size is fixed to 50. The number of iteration is fixed to 10.

Calculation of Fitness Value Each chromosome is consist of cl number of virtual machines where cl is the length of chromosome, i.e., equal to number of tasks and each VM is having its computing power or speed of execution. At a time cl tasks will be assigned to a chromosome, and the time taken to complete the execution of all tasks is the fitness value of that chromosome. So a chromosome with small

Fig. 3 Structure of chromosome



fitness value is the more powerful and likely to be selected.

$$FitnessValue = \sum_{i=1}^{cl} ET_i \tag{3}$$

Selection and Crossover Sort the chromosome with increasing order of their fitness value, and select the first m chromosomes and perform the crossover using the uniform crossover.

A simple example of uniform crossover is shown in Fig. 4. It means just a random exchange of elements (genes) between two chromosomes. In Fig. 4, G_1 and G_2 represent two chromosomes, and each number represents a gene. After crossover two new chromosomes O_3 and O_4 are generated.

Mutation A gene is selected randomly in a newly created chromosome and replaced with a randomly selected gene.

4.2 Experimental Results

To justify the significance of proposed cloud-edge centric architecture, we have performed our experiment by considering several service requests from different users in a smart city. The detailed parameter setting is as follows: The VM in the cloud layer is supposed to have more computing power and was set to 6000 to 9000 million instruction per second (MIPS). The VM in the edge layer is designed for executing small-size tasks to provide an immediate response to the users. So the computing power set in the range of 2000–6000 MIPS. Using Poisson distribution a real-time task generator was developed that distributes the inter-arrival time of task exponentially. The size of the task was set to range between 2000 and 12,000 million instruction (MI).

The proposed model is justified by considering three parameters: the percentage of tasks that completed their execution before the deadline, makespan, and throughput. For the first experiment, the number of VM was set to 300, 200 in the edge layer, and 100 in the cloud layer. The number of tasks was varied from 1000 to 5000. The comparison of the percentage of tasks that completed their execution before the deadline with the edge layer and without edge layer is shown in Fig. 5.

To justify the makespan, the number of VM was set to 300, and tasks were varied from 1000 to 5000. The experimental results show that the makespan is less in all five cases for cloud-edge architecture than without the edge layer. The result is shown in Fig. 6.

Fig. 4 Uniform crossover



We performed the third experiment by considering throughput of with and without edge layer. For calculating throughput, the number of tasks was fixed to 5000, and the number of VM was varied from 200 to 600. The comparison of throughput with and without the edge layer is shown in Fig. 7.

5 Conclusion and Future Work

As smart service provisioning in a smart city is highly demanding nowadays, proper utilization of the latest technologies and integration of these technologies is very essential. In this paper, we emphasize how to integrate cloud computing, edge computing, and the Internet of Things for smart service provisioning in a smart city. Internet of Things is used at the ground label to interconnect the IoT devices in a smart city. The edge computing has been used in between cloud and IoT layer to provide an immediate response to a service request. The performance of the proposed model was evaluated by considering the percentage of tasks that completed their execution, makespan, and throughput. The experimental result

Fig. 5 % of task completed with 300 VM

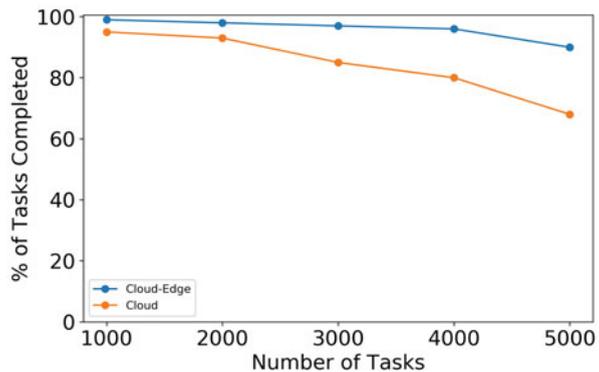


Fig. 6 Makespan with 300 VM

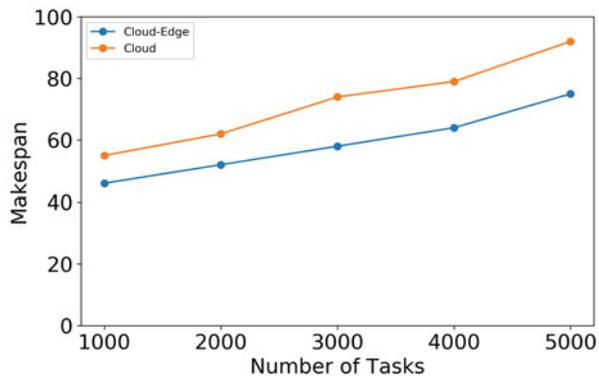
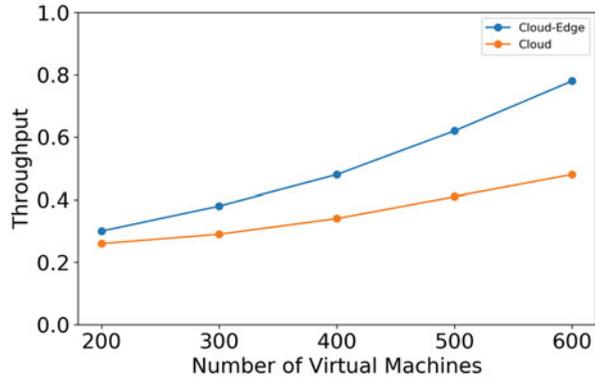


Fig. 7 Throughput with 5000 tasks



shows that the cloud-edge centric model performs better than without edge model. In the future, we will try to improve the model by applying some machine learning techniques.

References

1. D. Swati, M. Rajasekhara Babu, R. Patan, P. Jiao, K. Barri, A.H. Alavi, Internet of things based fog and cloud computing technology for smart traffic monitoring. *Internet of Things*, Elsevier **14**(100175) (2021)
2. W.H. Lee, C.Y. Chiu, Design and implementation of a smart traffic signal control system for smart city applications. *Sensors* **20**(2), 508 (2020)
3. D. Li, L. Deng, Z. Cai, Intelligent vehicle network system and smart city management based on genetic algorithms and image perception. *Mechanical Systems and Signal Processing*, Elsevier **141**, 106623 (2020)
4. S. Tian, W. Yang, J.M. Le Grange, P. Wang, W. Huang, Z. Ye, Smart healthcare: Making medical care more intelligent. *Global Health Journal*, Elsevier **3**(3), 62–65 (2019)
5. A. Kumar, Using cognition to resolve duplicacy issues in socially connected healthcare for smart cities. *Computer Communications*, Elsevier **152**, 272–281 (2020)
6. A. Subasi, L. Bandic, S.M. Qaisar, In *innovation in health informatics*, in *Cloud-Based Health Monitoring Framework Using Smart Sensors and Smartphone*, (Elsevier, 2020), pp. 217–350
7. B. Janakiramaiah, G. Kalyani, A. Jayalakshmi, Automatic alert generation in a surveillance systems for smart city environment using deep learning algorithm. *Evolutionary Intelligence*, Springer **14**(2), 635–642 (2021)
8. G. Baldoni, M. Melita, S. Micalizzi, C. Rametta, G. Schembra, A. Vassallo, In *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, in *A Dynamic, Plug-and-Play and Efficient Video Surveillance Platform for Smart Cities*, (Las Vegas, 2017), pp. 611–612
9. P. Bellavista, P. Chatzimisios, L. Foschini, M. Paradisioti, D. Scotece, In *2019 IEEE symposium on computers and communications (ISCC)*, in *A Support Infrastructure for Machine Learning at the Edge in Smart City Surveillance*, (Barcelona, 2019), pp. 1189–1194
10. A. Alelaiwi, A. Alghamdi, M. Shorfuzzaman, M. Rawashdeh, M. Shamim Hossain, G. Muhammad, Enhanced engineering education using smart class environment. *Computers in Human behavior*, Elsevier **51**, 852–856 (2015)

11. S.D. Nagowah, H. ben Sta, B.A. Gobin-Rahimbux. An Ontology for an IoT-Enabled Smart Classroom in a University Campus, In *2019 international conference on computational intelligence and knowledge economy (ICCIKE)*, Dubai, (2019), pp. 626–631
12. A. Pacheco, P. Cano, E. Flores, E. Trujillo, P. Marquez, in *2018 Congreso Internacional de Innovación y Tendencias en Ingeniería (CONIITI), IEEE, in A Smart Classroom Based on Deep Learning and Osmotic IoT Computing*, (2018), pp. 1–5
13. M.K. Patra, An architecture model for smart city using cognitive internet of things(CIoT). 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), 1–6 (2017). <https://doi.org/10.1109/ICECCT.2017.8117893>
14. S. Sahoo, B. Sahoo, A.K. Turuk, A learning automata-based scheduling for deadline sensitive task in the cloud. in *IEEE Transactions on Services Computing*. <https://doi.org/10.1109/TSC.2019.2906870>

Challenges for Swarm of UAV-Based Intelligence



Muhammed Akif Ağca, Peiman Alipour Sarvari, Sébastien Faye,
and Djamel Khadraoui

1 Introduction

Swarms of UAVs/drones are promising for many defence and safety-related applications and thus receiving more attention from industry and applied research institutions. Main services required include surveillance (air, land, maritime), package delivery (dangerous or not) and monitoring. In order to complete the required tasks, swarms of UAVs systems require cooperation and mission sharing in real time between each system unit.

Some applications use multi-agent-based autonomous information gathering via swarm of drones to help emergency teams [1]. Swarm intelligence is provided for target detection and tracking within a selected region via swarm of drones [2].

This paper covers the conceptual design of the main components of a system proposing a trusted holistic approach connected to the main architecture of swarms of UAVs system.

In order to relate our proposal to the literature, we present some existing work in relation to our concept design. From the computational side, fog computing-based approaches are emerging to enable task sharing among swarm units for mission/safety/operation-critical use cases. Uncertainty factors of harsh environments are considered in detail with fog computing facilities [3].

A distributed convex optimizer is proposed to speed up the optimization process and decrease latency in the context of swarm of drones [4]. However, increasing computational and storage capacities are decentralizing resources and algorithms.

M. A. Ağca (✉) · P. A. Sarvari · S. Faye · D. Khadraoui
Luxembourg Institute of Science and Technology – LIST, Esch-sur-Alzette, Luxembourg
e-mail: muhammed.agca@list.lu; peiman.alipour@list.lu; sebastien.faye@list.lu;
djamel.khadraoui@list.lu

In addition, swarming-based approaches enable more efficient task sharing, mission accomplishment and performance improvement for drones [5].

From the monitoring and control side, a learning-based anomaly detection and monitoring method for swarm drone flights is proposed in [6]. The indoor control of a swarm of drones in the context of an opera directed by a sound painter is also experimented in [7]. A low latency clustering method for large-scale drone swarms is proposed in [8].

From the existing state of the art, we may claim that the trusted scalability of the analytical functions and resources is still an open issue. In fact, edge devices, which have densified storage and computational facilities, enable a broader context and spectrum to be covered. Running novel machine learning algorithms at the edge by ensuring trust and security is a key enhancement. Nevertheless, the dynamic setting triggers the data/transaction flow in the system exponentially.

Valorizing the swarm intelligence and keeping the system resilient require real-time updates and predictions in different system layers. Ledger-based chain structures and big data technologies can accomplish the transaction scalability and memory speed analytic performance to some extent [9].

A recent paper by the authors [10] showed large-scale matrices generated by the novel methods having to be merged and fused dynamically to be able to scale/train the decentralizing algorithms. An increasing number of nodes in the system are causing swarm behaviour using sophisticated computational methods for the cooperative missions of autonomous system units.

If we look at the swarm of UAV/drone system from an architectural design point of view, we discover that the main challenging areas, in which improvements are still required, are trusted computational, control and communication system components. Despite this, mission/safety/operation-critical applications, such as tracking a moving object monitored by a swarm, require the trust to be verified at critical checkpoints while ensuring the performance of the system as a whole [10].

In this study, we propose a holistic approach to swarms of UAVs/drones' manipulation. It is based on an end-to-end trust mechanism, which involves monitoring and controlling the swarm of UAV/drone system in a trusted and scalable manner.

The paper is structured as follows. Section 2 shows the swarm of drones' system architecture with all its components. Section 3 explains the mechanism in details. Section 4 gives details about the trusted AI layer for swarm manipulation. Section 5 concludes the study and mentions future directions.

2 Swarms of UAVs/Drones System Architecture

2.1 System Architecture Trust Factors

Swarms of UAVs systems mainly have trusted computing, communication and control units. These have hard real-time constraints for task sharing and collaboration among system units.

The maximization of trust factors with a holistic view maximizes analytic edge capacities. The proposed architecture is based on an end-to-end trust mechanism, which ensures hard constraints in the data/transaction flow life cycle. The monitoring, detection and reaction of mechanisms' latency/throughput limits are optimized with a distributed checkpointing approach. This section gives details about the workflow in swarm of drones' system architecture.

For swarm data management, many innovations are proposed. Extending data locality to the edges in a trusted scalable manner enables the complexity of the data/transaction flow to be handled. It can keep the memory speed performance of analytical transactions in the swarm at a massive scale. The MEMCA (memory-centric-analytics) system models the number of node balancing problems proportional to the total throughput change, for any system. The holistic view based on the trust factor t enables dynamic monitoring and provisioning in a trusted scalable manner. It makes significant improvements to the USL (Universal Scalability Law) approach. Incidentally, the contention σ and crosstalk κ of any system can be minimized in order to maximize the throughput X of the system, as indicated in Eq. 1 [10]:

$$X(N) = \frac{\lambda \cdot N}{1 + \sigma(N - 1) + (1 - t) \cdot \kappa \cdot N \cdot (N - 1)} \quad (1)$$

Micro-service architecture has innovative approaches for layer-wise structures, enabling the verification of trust at critical checkpoints. Dynamic user-defined feature set management enabled at run time maximizes the performance of the cooperative missions. The trust factor of the resilient system requires a dynamic number of node N to be balanced for an optimal throughput X value. It requires the number of nodes N to be maximized for fluctuations in throughput changes. The derivative of Eq. 1 and equating to zero for local maxima gives the maximum number of nodes, as indicated in Eq. 2. Proposed approach can adapt the mechanism according to the throughput change of any system. It is improved with a feedback controller to train the system dynamically.

The holistic view approach enables end-to-end trusted scalable analytics by unifying resources. It can be extended to any other support in a trusted scalable manner with the checkpoint mechanism. Optimized checkpoint locations provide an efficient solution for the number of node balancing problems. However, latency requirements are dependent on remote resources. Additional layers are required to integrate other distributed and hybrid system designs.

$$N_{\max} = \sqrt{\frac{1 - \sigma}{\kappa \cdot (1 - t)}}. \quad (2)$$

The trust indicators considered in the MEMCA system are promising and will bring confidence to any network. They can be applied to predictions processed by the distributed ML algorithms on a massive scale. Meanwhile, smart ecosystems can implement them at any size as a core data fusion component for trusted scalable end-to-end analytics. This is a promising development for the end-to-end throughput requirements of blockchain structures. For instance, Hyperledger can achieve the end-to-end throughput of approximately 3500 transactions. It can scale well over 100 peers with sub-second latency [9]. It is useful for many use cases but not for the massive systems with larger data block sizes. Latency and bandwidth requirements have to be considered in detail in order to be able to maintain connectivity at such a large scale.

Emerging swarm systems require connected, cooperative and autonomous mobility. Every node has to be able to interact with any node in real time. Stand-alone units have limited resources. The system has to be supported with edge analytical facilities to provide semi-autonomy for the state-of-the-art methodologies. The critical factor is the enabling of a data/transaction flow between nodes in a trusted scalable manner.

Trusted scalable massive systems use emerging densified computation and storage units for their bottlenecks. The units improve the capacities of autonomous nodes. However, an increasing number of nodes exceed the scalability limits of current technologies, which are dependent on throughput fluctuations. Contention σ and crosstalk κ penalties increase as the diversity of system components increases, as demonstrated in Eq. 1. The connectivity of heterogeneous distributed systems has to be warranted to ensure the mobility of smart ecosystem components. Multi-agent-based mobility mechanisms can maintain connectivity in some manner and can be used to support many services, such as dynamic carpooling [11].

Security and trust can be ensured with service-oriented architecture (SOA). However, an increasing number of nodes, the diversity of system components and the decentralization of computing/storage facilities all require novel approaches to be able to maintain connectivity and support mobility services. Layer-wise structures and micro-service models are emerging to monitor and control swarm systems. Figure 1 illustrates the workflow in end-to-end trust mechanism.

2.2 Main System Architecture Components

Swarm of drones are controlled by hybrid designed (central, hierarchical, decentral) architectures. Trust indicators explained earlier in this section enable the verification of swarm intelligence in real time. This section explains the transaction workflow in swarms of UAVs/drones system, as illustrated in Fig. 1.

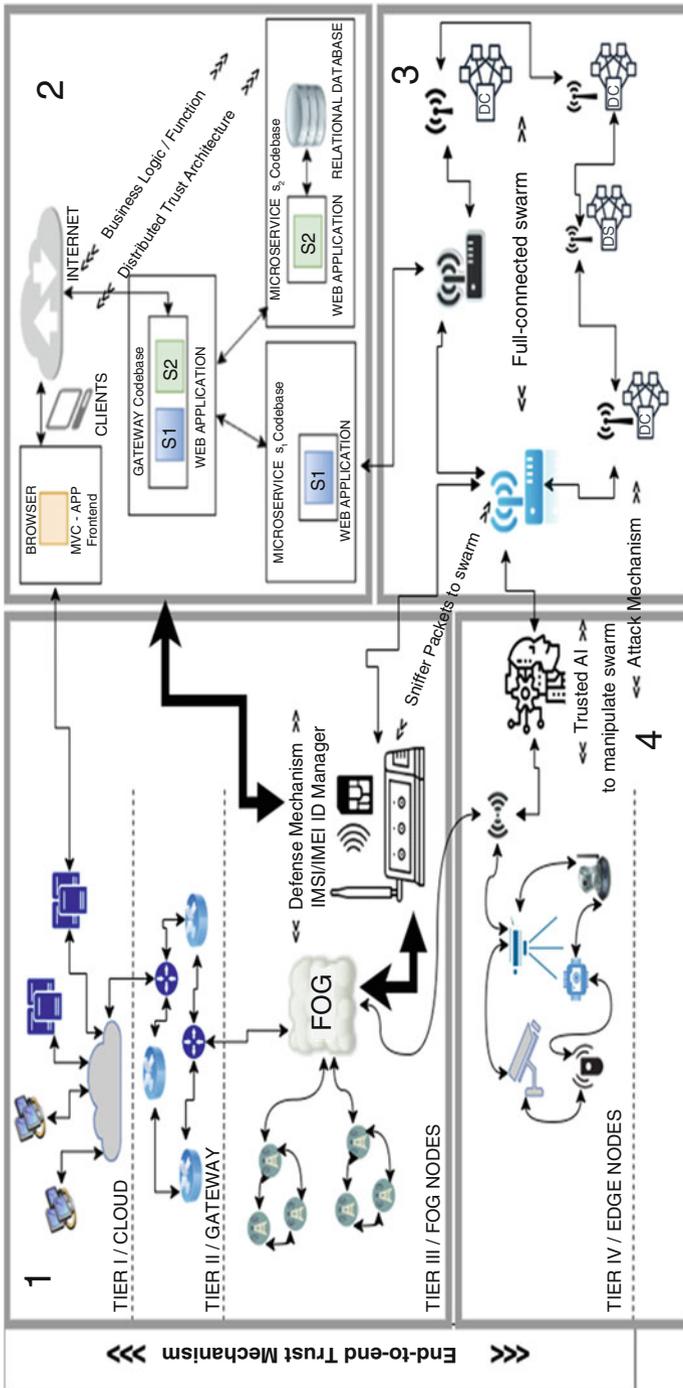


Fig. 1 End-to-end trust mechanism for swarm manipulation with trusted swarm intelligence

Firstly, the workflow is started with hierarchical MEMCA abstraction, an end-to-end trusted analytical code base for massive systems. It provides end-to-end, trusted analytics for swarm manipulation. It is composed of four tiers. The first tier, a core cloud-centric layer, has servers, storage and computational devices. The second tier is a gateway that supports the edge/fog layer interconnection and data transmission in a trusted manner. Smart sensor nodes are assigned unique IP addresses and controlled by cluster masters (CMs). The third tier is the fog layer, which consists of networking devices (routers, bases stations, etc.) with very low latency. The fog tier transmits the signals and connects the system with zero or near-zero latency. The fourth tier is composed by edge nodes, which interact with users and the environment. The edge tier collects datum up to requests from clients and monitors end-to-end transactions as data sensors.

Secondly, a distributed trust mechanism supports the transaction/data workflow. It builds and verifies end-to-end trust between the swarm units and provides trusted intelligence for decision mechanisms at a massive scale. Information access layers and user access categories are adapted dynamically. Trusted AI-based defence/attack mechanism adapt the layers and categories dynamically.

Thirdly, communication and vision-based swarm controllers interact with the large system via densified computing nodes. Networking protocols are dynamically adapted up to the computational context. IPV6 and TCP/UDP-based protocols are implemented dynamically for dynamic package broadcasting in a swarm. A distributed checkpointing mechanism-based verification approach ensures trust at all stages of the transaction life cycle via the embedded trust indicators.

Fourthly, trusted AI, designed for swarm manipulation, interacts with the IMSI/IMEI-based ID manager. Sections 3.1 and 3.2 explain the end-to-end trust mechanism and communication-/vision-based C2 (command control). Control architecture interacts with a dynamic feedback mechanism to train trusted AI knowledge bases. The load of each node is correlated dynamically with ΔX throughput fluctuations.

3 End-to-End Trust Mechanism

3.1 Trust Model on Distributed Agents for Swarms of UAVs

Trust models are mostly designed to be context-dependent and are not able to adapt dynamic changes during the run time. The changes for emerging end-to-end mechanisms have to be adjusted dynamically to keep track of the data/transaction flow. Secure multi-agent platforms can build trust to support mobility in some way [11].

Emerging architectures can persist the trust between the transactions at run time to adapt it to dynamic context changes in a trusted way. The end-to-end latency requirements can be satisfied with a distributed checkpointing mechanism [10].

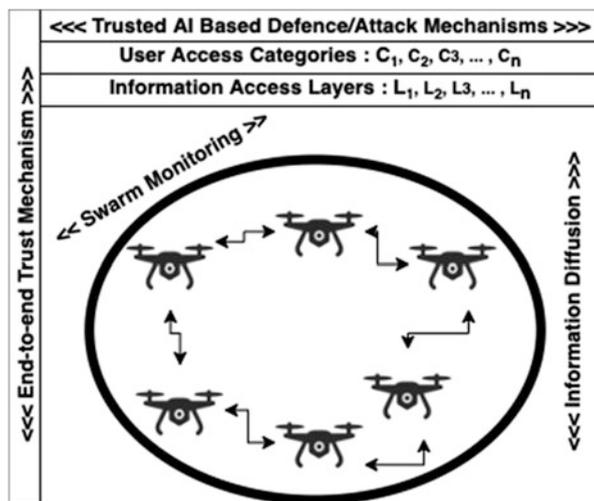
Dynamic optimization of checkpoint locations enables large systems to recover faster. It is supported by a micro-service design for swarm unit master nodes. Figure 2 illustrates the interaction of the end-to-end trust mechanism, which is between the swarm controller units of trusted AI system. The mechanism functions as a glue for each system unit to assure coherency in real time. Swarm monitoring transactions interact via agent sensors/actuators with the environment. Information fusion/diffusion strategies are updated dynamically based on the feedbacks from swarm manipulation protocols. User access categories and information access layers are dynamically managed to manipulate defence/attack mechanism. Section 3.2 is discussing about the C2 mechanism of the proposed intelligence system.

3.2 Communication- and Sensor-Based C2 for Swarms of UAVs

Interoperability is a de facto feature for mission/safety/operation-critical systems and must be ensured to enable real-time task sharing among swarm units. Interoperability challenges are presented in a live intelligence, surveillance and reconnaissance (ISR) experimentation. A high level of interoperability is successfully achieved in a joint ISR process, known as TCPED (tasking, collection, dissemination, exploitation and dissemination) [12].

The dissemination of data requires fusion/diffusion processes for data packages in a data streamflow. The latency limits of real-time systems make this challenging. However, emerging control mechanisms present novel solutions. Control problems are generally defined as manipulating the input with a defined set of processes to obtain the desired output from any system. Control strategies adapt to emerging

Fig. 2 High-level system architecture of trusted AI system with a defence/attack mechanism



systems. The decentralization of the tasks and total architecture is becoming popular.

In [13], the authors present the advantages of a distributed system rather than a hierarchical and a centralized one. Data fusion is implemented via a network of sensor nodes, each with a processing facility. Existing data fusion algorithms are criticized in terms of scalability and for wasting resources (communication and computation). Linear information filter is generalized for fully connected topologies and nonlinear systems. The system seems promising for solving a number of swarming task sharing issues among drones. However, limited computation and communication facilities in system units still require central communication for data-/CPU-bound tasks. A hybrid design of central, decentralized and hierarchical architectures with adaptive system units maximizes the total performance and throughput of swarm systems.

The uncertainty caused by harsh conditions can be eliminated with efficient control system modelling. Measuring sensitive parameters with sensor (visual and wireless) mechanisms is promising for control problem modelling and process manipulation for desired outputs. For instance, open-loop control is accomplished without any knowledge of current outputs [14]. A detailed explanation of our dynamic control and feedback mechanism is part of another study. This section reports on communication- and sensor-based C2 potentials and indicates major parts of our end-to-end trust mechanism-based concept design as illustrated in Fig. 1.

The holistic abstraction proposed in [15] enables whole transaction life cycle monitoring. It can maximize edge analytic facilities and minimizes bottlenecks/latencies in the whole system. The approach implements a feedback mechanism to train the system dynamically and minimize failures and recovery times.

New challenges are also emerging for networking in swarm systems because of decentralizing architectures. The IMSI/IMEI ID-based labelling of swarm system units is an efficient approach to be able to track each component. Any systems can be intercepted with IMSI/IMEI catcher/manager devices [16]. These can be implemented with lawful authorities or other technology owners. TCP/UDP, mobile and IPV6-based networking is applied to our concept design. Furthermore, fog layer-based communication is enabling new facilities. Lawful probe insertion/interception can be undertaken to track/monitor target regions or manipulate swarm traffic with IMSI/IMEI data. Interception for any band frequency (VoIP gateway, 1G/2G/3G/4G/5G, GPRS) can be carried out in a fog layer within the latency requirements of mission/safety/operation-critical systems. Sniffing attacks for swarm manipulation are managed in the fog layer.

The C2 mechanism for swarm systems is supported with a vision-based coordination unit, densified computing nodes and densified storage nodes. The architecture enables package broadcasting and data fusion/diffusion in real time or near real time. Section 4 gives details about trusted AI-based swarm manipulation approaches and the advantages of our concept design.

4 Trusted AI for Swarm Manipulation

Trust development is an attributional process, and perceived trust is an essential aspect of developing and maintaining interpersonal relationships. Successful cooperation between human communicators occur when ambiguity and uncertainty in social perceptions are reduced through the development of trust [17].

Besides the issues in software engineering, a key challenge is meeting user’s expectations with consistent system performance. As far as the performance of AI systems is concerned, trust issues cannot be ignored. Trust is gained by understanding the reasoning behind an AI system’s conclusions and results, which is more about being confident [18] than having explicable in the accuracy.

Trusted AI systems are considered and implemented for mission/safety/operation-critical systems [19, 20]. The state-of-the-art AI platform is trained and tested via the data from available edge nodes for a particular and fully connected swarm on top of a MEMCA abstraction. The training data is the labelled dataset from fully connected edge nodes and swarm devices. The fitted model empowers the system to handle all input nodes despite the limitations of blockchain technology, which is restricted by scalability and latency factors. This AI model will be embedded in the verification tier of the platform by manipulating swarm to obtain the highest levels of trustfulness and resiliency.

Figure 3 illustrates the high-level framework of the proposed AI platform that trains and tests the IoT data. This phase needs labelled input from swarm via sniffer

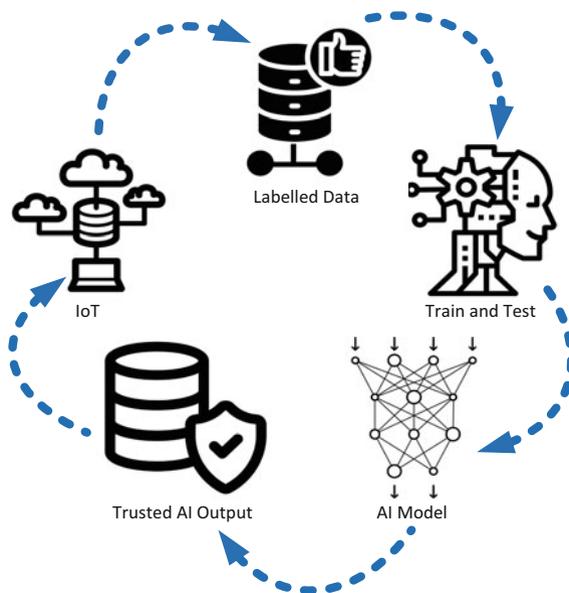


Fig. 3 The high-level AI platform to produce swarm manipulation protocol

packets. The trusted AI output consists of the practical manipulation metrics (with a specific hard and soft window based on the significant model error level) for ensuring the resiliency and trustfulness of the system in a scalable manner.

Automation, industrialization, self-regulation and motorization with built-in AI is a developing field with different uses in numerous industries. [21] contributes to a deeper comprehension of how businesses can foster trust in practical AI. Trust can be used as a particular construct for the permission by shedding light on the driving technologies to establish trust in the technology, as well as confidence in the innovating firm and its language. From another point of view, the strong traceability power of blockchain technologies with ledger-based chained structures might be the missing piece of the confusing and bewildering puzzle for advanced, granted and trusted data governance and data provenance for AI model failure diagnoses [22, 23].

Practitioners in the field of AI are very concerned about the trustworthiness of their built models and their acceptability by human users. [24] claim that awareness of the rich set of solutions developed in the multi-agent systems should be considered a subfield of trust modelling. [25] describes a structure and skeleton for handling AI use, as well as its key elements. They also reveal detailed patterns that show how this structure facilitates the management and performance of model training and connected learning pipelines while introducing trusted AI laws.

Generally speaking, trust-based mechanisms have five fundamental components: monitoring and information gathering, a trust calculation and evaluation unit, a trust recommendation unit, decision-making and the diffusion of the detection unit. Nowadays, a trust-based strategy renders a reputation scheme composed of different components to provide security against routing attacks. [26] implements an in-depth examination of numerous elements in the trust-based tools and procedures used for the practical and useful accomplishment of a task by looking at the details.

Detecting anomalies and misbehaviours of agents/drones in the system is possible within the trust development/monitoring process. Online/offline collision detection/assessment and stochastic models that assess collision can be implemented for a task defined in the swarm. However, end-to-end latency requirements and hard constraints of the swarm drones' systems limit the edge analytic facilities. Trusted scalable holistic abstraction to automate the life cycle of AI transactions can minimize latency in bottlenecks.

The distributed checkpointing mechanism proposed in [15] allows the verification of trust for mission/safety/operation-critical applications with an optimal total performance. We will be sharing the experimental results of our end-to-end trust mechanism-based AI system in another study. Initial reflections of our vision show promising solution approaches to the challenges of swarms of UAVs/drones. Section 5 concludes the study and briefs on future opportunities.

5 Cross-Border Challenges

One of the use cases that have a growing potential in the literature involves cross-border areas, which mostly have critical needs and yet many connectivity and coverage problems.

One example of this is the France-Luxembourg cross-border area, where thousands of drivers commute every day, creating some of the most important traffic jams in the region, leading to loss of time and a significant economic and environmental impact.

More and more mobility applications are being developed in an effort to alleviate the situation. These applications need connectivity, either distributed (e.g. IEEE 802.11p) or centralized/hybrid (e.g. 5G). UAV/drone swarms can provide a solution to meet specific latency and communication needs by integrating them as relay or control nodes, in addition to conventional communication nodes (e.g. antennas, satellite, etc.). Their mobile capability makes it possible to dynamically adjust the network coverage according to the needs (in latency, throughput, etc.). In the same way, due to their mobile capability, UAVs can be used in specific assistance cases to track, for example, ambulances or security/police vehicles.

Whatever the case, the UAVs described in these scenarios will have an irreplaceable need for reliability and trust, which will have to be applied to every possible layer of their architecture in order to ensure that the information transiting through them is delivered with all the necessary conditions for the proper execution of the applications.

6 Conclusion

We constructed an investigation to reflect our initial vision for state-of-the-art challenges on swarms of UAVs/drone-based intelligence and potential solution approaches. Moreover, we elaborated that the methodologies can assure trustworthiness/security against swarm manipulation benefiting AI in the edge in a trusted scalable manner.

Big data technologies and ledger-based chained structures are innovative enough to overcome scalability concerns and memory speed limitations. A solution approach has been presented in this study, which was adapted to the case of swarms of UAVs/drones.

This paper gives ideas for future work, such as scrutinizing sophisticated mathematical modelling-aided ML algorithms for the swarm-based structures in order to comply with scalability and obtain more accurate estimates of module execution time. The models enable the whole AI life cycle to be automated and assure trust for monitoring/detect/react mechanisms within the constraints of mission/safety/operation-critical systems. Mobility and connectivity simulation scenarios (VANETs) for the trusted AI system and swarm intelligence mechanism are promising for initial simulation experiments.

References

1. A. Viseras, T. Wiedemann, C. Manss, V. Karolj, D. Shutin, J. Marchal, Beehive-inspired information gathering with a swarm of autonomous drones. *Sensors* **19**(19), 4349 (2019)
2. M.G. Cimino, M. Lega, M. Monaco, G. Vaglini, Adaptive exploration of a UAVs swarm for distributed targets detection and tracking, in *ICPRAM*, (2019, February), pp. 837–844
3. X. Hou, Z. Ren, J.Wang, S. Zheng, W. Cheng, H. Zhang, Distributed fog computing for latency and reliability guaranteed swarm of drones. *IEEE Access* **8**, 7117–7130 (2020)
4. X. Hou, Z. Ren, W. Cheng, C. Chen, H. Zhang, Fog based computation offloading for swarm of drones, in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, (IEEE, 2019, May), pp. 1–7
5. A. Mirzaeinia, M. Hassanalilian, K. Lee, M. Mirzaeinia, Performance enhancement and load balancing of swarming drones through position reconfiguration, in *AIAA Aviation 2019 Forum*, 2019, p. 3463
6. H. Ahn, H.L. Choi, M. Kang, S. Moon, Learning-based anomaly detection and monitoring for swarm drone flights. *Appl. Sci.* **9**(24), 5477 (2019)
7. S. Chaumette, D.A.G. Jáuregui, S. Bottecchia, N. Couture, Issues of indoor control of a swarm of drones in the context of an opera directed by a Soundpainter (May 2019)
8. X. Zhu, C. Bian, Y. Chen, S. Chen, A low latency clustering method for large-scale drone swarms. *IEEE Access* **7**, 186260–186267 (2019)
9. E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, et al., Hyperledger fabric: A distributed operating system for permissioned blockchains, in *Proceedings of the Thirteenth EuroSys Conference*, (ACM, 2018), p. 30
10. M.A. Ağca, D. Khadraoui, S. Faye, Persisting Trust in Untrusted Varying Resilient City Context V, in *Proceedings on the International Conference on Artificial Intelligence (ICAL)*, (The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2019), pp. 312–318
11. C. Bonhomme, G. Arnould, D. Khadraoui, Dynamic carpooling mobility services based on secure multi-agent platform, in *2012 Global Information Infrastructure and Networking Symposium (GIIS)*, (IEEE, 2012), pp. 1–6
12. R. Vicen-Bueno, G. Cimino, D. Cecchi, B. Garau, Live tasking/command and control (C2) of ISR unmanned underwater gliders from remote operational sites, in *Oceans 2019 MTS/IEEE Seattle*, (IEEE, 2019), pp. 1–12
13. A.G. Mutambara, *Decentralized Estimation and Control for Multisensor Systems* (Routledge, 2019)
14. R.G. Jacquot, *Modern Digital Control Systems* (Routledge, 2019)
15. M.A. Ağca, A holistic abstraction to ensure trusted scaling and memory speed trusted analytics, in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, (IEEE, 2019, December), pp. 1428–1434
16. S. Park, A. Shaik, R. Borgaonkar, J.P. Seifert, Anatomy of commercial IMSI catchers and detectors, in *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society*, (2019), pp. 74–86
17. J. Hohenstein, M. Jung, AI as a moral crumple zone: The Effects of AI-mediated communication on attribution and trust. *Comput. Hum. Behav.* **106**, 106190 (2019)
18. L. Ding, Human knowledge in constructing AI systems—Neural logic networks approach towards an explainable AI. *Procedia Comput. Sci.* **126**, 1561–1570 (2018)
19. B.G. Kang, K.M. Seo, T.G. Kim, Machine learning-based discrete event dynamic surrogate model of communication systems for simulating the command, control, and communication system of systems. *Simulation* **95**(8), 673–691 (2019). 0037549718809890
20. J. Falcone, Machine learning systems in nuclear command, control, and communications architecture: Opportunities, limitations, and recommendations for strategic commanders. *NAVAL POSTGRADUATE SCHOOL MONTEREY CA MONTEREY United States* (2019)

21. M. Hengstler, E. Enkel, S. Duelli, Applied artificial intelligence and trust—The case of autonomous vehicles and medical assistance devices. *Technol. Forecast. Soc. Chang.* **105**, 105–120 (2016)
22. K. Sarpatwar, R. Vaculin, H. Min, G. Su, T. Heath, G. Ganapavarapu, D. Dillenberger, Towards enabling trusted artificial intelligence via blockchain, in *Policy-Based Autonomic Data Governance*, (Springer, Cham, 2019), pp. 137–153
23. M. Nassar, K. Salah, ur Rehman, M. H., & Svetinovic, D., Blockchain for explainable and trustworthy artificial intelligence. *Wiley Interdiscip. Rev. Data Min. Knowl. Disc.* **10**(1), e1340 (2020)
24. R. Cohen, M. Schaekermann, S. Liu, M. Cormier, Trusted AI and the contribution of trust modeling in multiagent systems, in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, (International Foundation for Autonomous Agents and Multiagent Systems, 2019, May), pp. 1644–1648
25. W. Hummer, V. Muthusamy, T. Rausch, P. Dube, K. El Maghraoui, A. Murthi, P. Oum, Modelops: Cloud-based lifecycle management for reliable and trusted AI, in *2019 IEEE International Conference on Cloud Engineering (IC2E)*, (IEEE, 2019), pp. 113–120
26. N. Khanna, M. Sachdeva, Study of trust-based mechanism and its component model in MANET: Current research state, issues, and future recommendation. *Int. J. Commun. Syst.* **32**(12), e4012 (2019)

Contrived and Remediated GPU Thread Divergence Using a Flattening Technique



Lucas Vespa and Genevieve Peters

1 Introduction

Branch divergence can severely degrade GPGPU performance [1]. Although many methods have been proposed to mitigate this problem, highly divergent branch-based code still limits how much these methods can help. In this work, we show how to modify divergent source code in order to create divergent free code. We demonstrate this by arranging extremely divergent code that would normally be ruled out for execution on GPU. We then apply our method, which eliminates divergence completely.

Here is why it works. Branches can cause performance issues in many systems, which is why a great deal of work has been done to try and mitigate their effects. Unlike multiprocessors with 8, 16, or 32 cores, GPUs are massively parallel with thousands of processing elements. This gives us more leeway in how we deal with branches on GPU. The best GPU applications have no branches and that is what we do in this work – we eliminate all branches. This is drastically different from previous work. The way we eliminate branches seems counterproductive. It looks at first glance as if we are creating less efficient code. That is because the code is less efficient, on CPU. But on GPU, with many processors, the code runs faster than its branched equivalent.

In this work we present an often extreme de-optimization for CPU which completely eliminates branches and results in a significant optimization for general-purpose GPU and SIMD applications, especially applications which had previously not been suitable for GPU implementation due to thread divergence. Our optimization removes all branches from code blocks and replaces each block with

L. Vespa (✉) · G. Peters
University of Illinois, One University Plaza, Springfield, IL, USA
e-mail: lvesp2@uis.edu; gpete2@uis.edu

a reduced equation. The equation evaluates all branches simultaneously using arithmetic operations. We call our method algorithm flattening (AF). Utilization of AF eliminates thread divergence and substantially decreases execution time from an already GPU accelerated state. Algorithms previously unsuited for GPU acceleration can be implemented on GPU with AF. Deterministic performance is achieved, and static instruction scheduling can be used (if desired).

Essentially, we destroy GPU performance as much as branch divergence can destroy it. Then we show that AF can revive it. We do so not only to show the potential of AF but also to show that AF can open the door for non-GPU-compatible code (i.e., highly divergent code), to potentially be run more successfully on GPU.

The remainder of this work is organized as follows. Section 2 presents background information on GPU architecture and considerations and related work in the area of branch elimination. Our work on branch elimination is detailed in Sect. 3. Results are presented in Sect. 4 and the paper is concluded in Sect. 5.

2 Background and Related Work

A general GPU architecture consists of multiprocessors (MP) or blocks containing individual processing units (PUs) running threads in an SIMD arrangement [2, 3]. Threads in the same block can generate max performance when all processing units start and finish assigned jobs at the same time, and no single PU ever executes an instruction with NULL values. Control divergence means some threads complete the jobs assigned to them but still need to wait for other threads to finish their jobs before they can work on new assignments; or more commonly, that some PUs are idle, while others in the same MP execute a different branch. The biggest contributor to control divergence in any GPU kernel is branch code.

Many techniques have been proposed to reduce the impact of branches. Some of these techniques are more general and suitable for any platform, while others are specifically designed for SIMD. The most common techniques involve analyzing [1, 4] and merging conditions [5–7], reordering conditions [8], replicating code or code addition [9], factoring code [10], distributing branch code [11–13], hardware-based predicated execution [14, 15], and warp scheduling [16, 17]. These techniques have the same end goal as we do, and some of these techniques are actually orthogonal to ours, making them suitable for simultaneous implementation with our optimization.

Anido et al. [18] reduce the number of branches executed using guarded instructions and pseudo-branches, which selectively execute instructions based on register values. Pseudo-branch instructions require an additional tag that allows control to be skipped to a destination point, so the SIMD device must have the appropriate hardware. Carrillo et al. [11] describe two optimizations for general-purpose GPU using code splitting. Loop splitting takes a loop and breaks it into two or more smaller loops. Branch splitting involves breaking segments of a branch off into separate kernels. The purpose of splitting code is to reduce the hardware load. For data-dependent loops, Sarkar and Mitra [13] split code by discovering if

a significant pattern exists in the control flow over a period of iterations. Branch behavior determines where code is split and how many kernels result. These methods are generally orthogonal to ours but like ours will result in de-optimization for CPU execution. Han and Abdelrahman [10] present an optimization method known as iteration delaying. Similar to guarded instructions and branch splitting, segments of branches are gated. Unlike the other methods, iteration delaying delays segments from executing so that similar segments are executed together.

Code replication, an optimization technique described by Mueller and Whalley [9], simplifies control flow by replicating segments of code. Han and Abdelrahman [10] also present an optimization method known as branch distribution in which common code between branches is replicated outside of the branches. Branch distribution and code replication are similar to our work; new code is added which creates an optimization for SIMD but a de-optimization for CPU. However, these methods simply reduce the time penalty due to thread divergence but do not actually eliminate branches.

Chakroun et al. [19] reduce thread divergence in a branch and bound algorithm accelerated on a GPU. As part of their method, they remove several single-level if-else statements through a method equivalent to α (Eq. 1) for case $n=0$ only. This alone is a significant contribution. This paper does not extend this method from 2012 [19]. Rather, we extend the multi-level algorithm flattening from 2011 [20] in Section II-F.

Other methods of SIMD optimization such as hardware predicated execution [14] can be used to reduce the number of branch operations executed in a program but require run-time intervention. Reissmann et al. [15] implement a predicated branch restructuring algorithm for unstructured code. Yu et al. [17] redesign PDOM stack hardware and implement a multi-level scheduling protocol to run more warps at once and re-converge threads earlier. Rogers et al. [16] offer a hardware solution that profiles memory accesses and dynamically reschedules warps to reduce the frequency of re-referencing data. Rogers et al. [16] note that their method is not intended to solve divergence but is a viable option in the ongoing effort to optimize GPU code.

Fung et al. [12] attempt to improve the efficiency of how branches are executed by regrouping threads. Liang et al. [21] demonstrate that performance gains from thread regrouping are more accurately measured with thread modeling and basic block vector metrics. This in-depth analysis of thread-level control flow divergence can guide optimization strategies. In support of general-purpose GPU optimization techniques, Yu et al. [22] compare threads by calculating the degree of similarity in their execution paths and represent these values on a grayscale. Yu et al. [22] visualize this evaluation as a graphed matrix where thread divergence is easily identified from the variability in tones.

Lin et al. [23] assess divergence at run time and implement a thread-data remapping algorithm to reduce global memory accesses. Thread-data remapping avoids source code optimization and does not eliminate branches. For specific programs where random selection determines control flow, like the fractal flames algorithm, Schied et al. [24] propose randomizing the data instead to achieve

intra-warp synchronicity, thus eliminating branches. Huang and Yang [25] redesign parallel loops so that idle threads can execute ensuing iterations tasked to non-idle threads.

Branch fusion [1] optimizes code by “weaving” together divergent branches with similarities. Multiple branches can also be merged together through conditional merging [5–7], reducing the total number of paths. This technique trades precision for performance and is recommended for error-tolerant applications [7]. Reordering of branches may also result in increased performance in the average case [8]. Branch reordering is a general optimization suitable for CPU or GPU implementation and is orthogonal to our work.

Unlike most optimization techniques, Grigorian and Reinman [26] offer a neural network solution that automatically identifies divergent kernels, trains artificial neural networks, and approximates target kernels with branch-less code. Although branches are eliminated, this method can be intensive and produce only approximate results.

While these techniques can help reduce the penalty of branch divergence and processing branches, none of them completely eliminate this penalty, because none of them completely eliminate branches without a trade-off of approximation. Therefore, using these techniques, there remains overhead in processing branch code, as well as stalling of SIMD processing elements due to divergent threads. Our optimization completely eliminates divergent code and therefore all overhead associated with handling branches and divergent threads.

3 Branch Elimination

By using a method called algorithm flattening (AF), we are able to replace branch statements with equivalent mathematical expressions that GPUs can process more efficiently. As a preview of how AF works, observe Eq. 2, which is the flattened version of the code in Fig. 4.

3.1 Applying Preliminary AF to Simple Branches

The basic idea behind algorithm flattening is to represent an entire branch with a mathematical expression, beginning with a simple non-optimized flattening process. A basic flattened equation results from the summation of the product of each assignment and its corresponding evaluated expression. The result is that one expression can be executed that represents the whole branch. For example, in Fig. 1, an if statement for variable reassignment can be represented by the expression

$$x = (e)p + (!e)x$$

Fig. 1 Basic if statement

```
if (e) {
    x = p;
}
```

Fig. 2 Basic if-else statement

```
if (e) {
    x = p;
} else {
    x = q;
}
```

Fig. 3 Basic nested if-else statement

```
if (e1) {
    if (e2) {
        x = p;
    } else {
        x = q;
    }
} else {
    x = r;
}
```

where e represents the expression being evaluated, p is the assignment value if e is true, and x is the assignment value when e is false. When the expression is evaluated, two outputs are possible: $x = p$ or x is unchanged ($x = x$).

if e is true,

$$x = (1)p + (0)x \longrightarrow x = p$$

and if e is false,

$$x = (0)p + (1)x \longrightarrow x = x$$

More complex branches such as if-else, nested if-else statements and chains, switches, etc. can also be generalized. If-else branches are similar to if statements, as seen in Fig. 2. However, we must apply the else statement assignment as well. So the if-else expression would read

$$x = (e)p + (!e)q$$

The flattening of a nested if-else statement into a mathematical expression requires substitution. For example, the nested if-else condition described in Fig. 3 can be flattened by first splitting the code into two distinct parts. The first part is the inner or nested if-else condition. Once we convert the entire nested part of the code, we can treat it as an outcome for the parent conditional. Then flattening the remaining parent conditional is the same process as before, with the added step of substituting in the nested conditional as one of the parent conditional’s specific outcomes. This can of course be automated recursively. The AF result for Fig. 3 is as follows:

$$x = e1((e2)p + (!e2)q) + (!e1)r$$

3.2 Generalized Preliminary AF for All Branches (A Starting Point)

$$\alpha = \sum_{i=0}^n x_i \cdot y_i = (x_0 \cdot y_0 + x_1 \cdot y_1 + \dots x_n \cdot y_n)$$

$$\beta = \prod_{i=0}^n x_i \cdot y_i = (x_0 \cdot y_0 \cdot x_1 \cdot y_1 \cdot \dots x_n \cdot y_n) \quad (1)$$

where $y_i = \alpha \vee \beta \vee \text{assigned_value}$

Equation 1 shows the non-optimized and unreduced, generalized format for performing AF on any code block. The finalized AF equation is either a sum or product with other sums and products embedded. A pseudo-algebraic reduction, as well as other optimizations, makes AF much more efficient as shown shortly.

3.3 Optimized and Reduced AF

The most common instruction in an AF-reduced equation is multiply-add. A compute 3.5 capable NVIDIA GPU can execute 192 multiply or multiply-add operations in one cycle. AF also reduces instructions because of the multiply-add operation within GPUs which allows multiple parts of the expression to be evaluated in one cycle. Also, flattened expressions can reduce instructions by omitting redundant variable assignments, namely, assignment to zero. Algebraic reduction is another method to further reduce an AF expression. Other possibilities for reduction exist in many cases. For example, it is known that

$$(e)a + (!e)b = e(a - b) + b$$

which is often reduced further if constants, multiples, or other factors reduce algebraically. Also, branch reordering can be used to reduce AF equations. This is in contrast to standard forms of branch reordering which concentrate on reducing the average path through branched code. This intuitive kind of optimization makes no difference for the performance of an AF equation. Reordering to reduce AF involves choosing an order which allows an AF equation to fully reduce. Branch reordering, which in an if-else statement actually involves inverting the conditional expression, can be represented as follows:

$$e(a - b) + b = !e(b - a) + a$$

It should be noted that reordering and inversion can often be achieved algebraically without knowing the above reordering equality by recognizing any variation of the following equivalence:

$$-e + 1 \equiv !e$$

This is demonstrated in the first form of reduction in Sect. 3.4.

3.4 A Worked Example

The following is a worked example of how to convert a conditional statement represented in code into a mathematical expression, with some optimization. Figure 4 shows the example, which is not contrived but rather derived from [27]. This first step of conversion is to identify the nested conditionals as follows:

$$if(m == 1) \{x = b; \} else \{x = b + 1; \}$$

Initial flattening uses Eq. 1

$$x = (m == 1) \cdot b + (m \neq 1) \cdot (b + 1)$$

which is reduced as follows using $[e(a-b)+b]$:

$$\begin{aligned} x &= (m == 1) \cdot (b - (b + 1)) + b + 1 \equiv \\ x &= (m == 1) \cdot (-1) + b + 1 \equiv \\ x &= [-(m == 1) + 1] + b \equiv \quad \text{Note : } -e + 1 \equiv !e \\ x &= [!(m == 1)] + b \equiv \\ x &= (m \neq 1) + b \end{aligned}$$

This is simplified quicker with branch reordering using $[!e(b-a)+a]$ rather than $[e(a-b)+b]$:

$$\begin{aligned} x &= (m \neq 1) \cdot (b + 1 - b) + b \equiv \\ x &= (m \neq 1) \cdot (1) + b \equiv \\ x &= (m \neq 1) + b \end{aligned}$$

After nested control flow is flattened, parent conditionals are flattened while substituting the converted nested conditional for the parent conditional’s outcome. This leads to the basic AF equation

$$x = (n == 0) \cdot a + (n == 1) \cdot (a + 1) + (n == 2) \cdot ((m \neq 1) + b)$$

Which is simplified to Eq. 2.

$$x = !(n >> 1) \cdot (n + a) + (n == 2) \cdot ((m \neq 1) + b) \quad (2)$$

4 Results

All experiments are performed using an NVIDIA TESLA K40 GPU, and results are gathered using CUDA C and NVIDIA NSIGHT. Our results are aimed at demonstrating what happens when using AF with different amounts of divergence. We therefore have contrived structured code with varying numbers of branches, as well as varying input. These are the two biggest factors: divergence and input. If divergence is great, it can have an extremely negative impact on performance. However, if the code has potential divergence issues, the input data can either calm or exacerbate this divergence, as we demonstrate here.

Specifically, we modify the code from Fig. 4 to have a varying number of branches. We then provide input data consisting of random values within a varying range. The input data guides how much divergence is actually seen during code execution, which affects performance, as can be seen in the following results.

Figure 5 shows the execution time of the branched algorithm with 8 branches and its AF equivalent. The range of random values for n and m varies from 8 to a maximum possible of 128. AF outperforms the branched code regardless of the range of input values. However, the speedup of AF diminishes as the range of input values increases due to the non-deterministic performance of the branched code. This is because, as the range of input values increases, thread divergence in the branched code decreases to a certain point, as the branch path taken for each processing unit is more unified. In addition to the speedup AF achieves, AF also exhibits completely deterministic performance, regardless of input. This potentially means a great deal to GPU architects, as well as security architects, as deterministic performance is a key concern in the development of these areas. Interestingly, the algorithm is only designed to take a range of eight input values. So although this

Fig. 4 Reassignment example. The finalized AF version of this example is shown in Eq. 2

```

if (n==0) {
    x = a;
}
else if (n==1) {
    x = a+1;
}
else if (n==2) {
    if (m==1) {
        x = b;
    } else {
        x = b+1;
    }
}
else {
    x = 0;
}

```

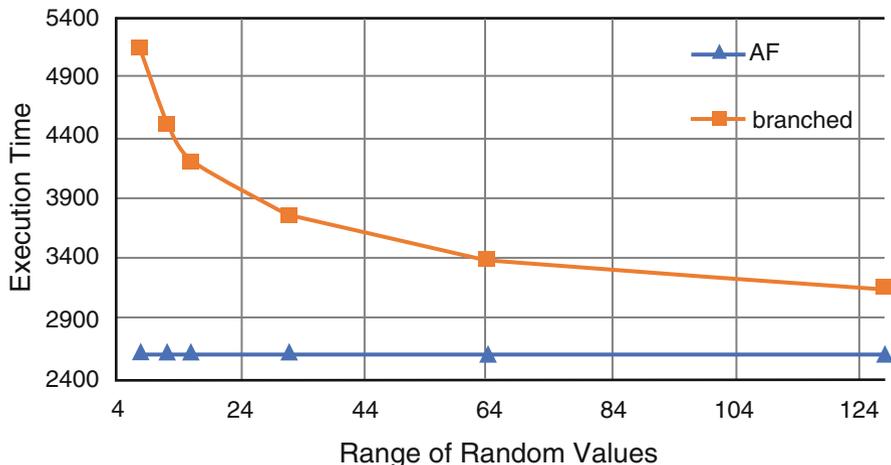


Fig. 5 Execution time of code with eight branches and varying data values, including original branched code and the flattened (AF) equation

graph shows an increase in performance with an increase of random value range (branched code), this would not be seen in practice and is simply for demonstration purposes only. One more thing to consider is that if the range of random values falls below 8, this could potentially swing the performance in favor of a branched algorithm. This is demonstrated in a later result. If the data is consistent and deterministic, this can outweigh the fully deterministic property of AF, which comes at the cost of increased instruction count.

Figure 6 shows the execution time of the branched algorithm with 64 branches and its AF equivalent. The range of random values for n and m varies from 8 to a maximum possible of 128. Figure 6 demonstrates the control that data input can have on divergence. In Fig. 6, AF outperforms its branched equivalent substantially under average case input (64 different random values). However, with outside of average or rare circumstances, the performance of the branched code nears and even crosses over the performance of AF. This is seen when the range of random values is low but the number of branches is high (64 in this case), mirroring what we see in Fig. 5. This unifies thread execution between threads, decreasing the effect of divergence. This also begins to happen when the number of possible input values is much larger than the 64 possible branch decisions. However, in the average case, AF performs excellent compared to its branched counterpart.

A more straightforward explanation of Fig. 6 might be as follows. If there are 64 random values, divergence is maximized, in that each thread has a lower probability of executing the same branch as another thread. In other words, if there is only one possible input value, every thread would take the same branch and execute code from that branch. If there are two possible values, approximately every other thread would take the same branch, so divergence would increase from the single value

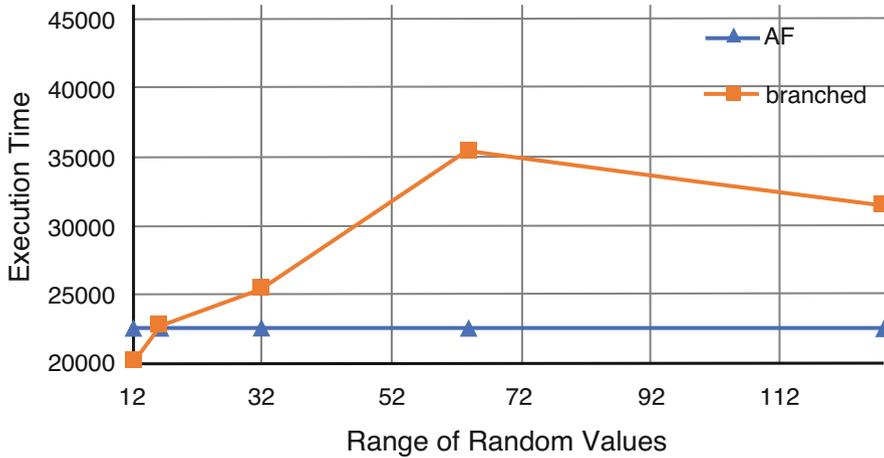


Fig. 6 Execution time of code with 64 branches and varying data values, including original branched code and the flattened (AF) equation

scenario. However, after 64 random values, the threads begin to slowly unify again because any value in excess of 64 will execute the same branch. Therefore, after 64 random values, with 64 branches, performance increases.

5 Discussion and Conclusion

SIMD design results in the most processing elements per area and therefore will always be able to push the limits of parallelization and performance, regardless of technology. It is therefore important for GPU research to integrate programming and compiler methods such as AF to get the most out of the capability of GPU architectures, which seem to be the future of compute-intensive applications.

AF has been shown in this work to improve performance in situations of divergence, especially when data exploits divergent code. When potential divergence exists in code for GPU, data typical to an application can easily exploit this divergence. However, in this work we show that there is a range of divergence caused by varying input data, and AF is a potential solution to flatten or average out the effect of this range of divergence.

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

References

1. B. Coutinho, D. Sampaio, F. Pereira, W. Meira, Divergence analysis and optimizations, in *Parallel Architectures and Compilation Techniques (PACT), 2011 International Conference on*, Oct 2011, pp. 320–329
2. M. Flynn, Some computer organizations and their effectiveness. *IEEE Trans. Comput.* **C-21**(9), 948–960 (1972)
3. W. Zhang, T. Bao, B. Zang, C. Zhu, Data pipeline optimization for shared memory multiple-simd architecture, in *Proceedings of the 19th International Conference on Languages and Compilers for Parallel Computing*, 2007, pp. 49–63
4. R. Bodík, R. Gupta, M.L. Soffa, Interprocedural conditional branch elimination, in *Proceedings of the ACM SIGPLAN 1997 Conference on Programming Language Design and Implementation*, 1997, pp. 146–158
5. W. Krehling, D. Whalley, M. Bailey, X. Yuan, G.-R. Uh, R. Engelen, Branch elimination via multi-variable condition merging, in *Euro-Par 2003 Parallel Processing*, ser. Lecture Notes in Computer Science, vol. 2790 (Springer, Berlin/Heidelberg, 2003), pp. 261–270
6. W.C. Krehling, D. Whalley, M.W. Bailey, X. Yuan, G.-R. Uh, R. van Engelen, Branch elimination by condition merging. *Softw. Pract. Exp.* **35**(1), 51–74 (2005)
7. J. Sartori, R. Kumar, Branch and data herding: Reducing control and memory divergence for error-tolerant GPU applications. *IEEE Trans. Multimedia* **15**(2), 279–290 (2013)
8. M. Yang, G.-R. Uh, D.B. Whalley, Improving performance by branch reordering, in *Proceedings of the ACM SIGPLAN 1998 Conference on Programming Language Design and Implementation*, 1998, pp. 130–141
9. F. Mueller, D.B. Whalley, Avoiding conditional branches by code replication, in *Proceedings of the ACM SIGPLAN 1995 Conference on Programming Language Design and Implementation*, 1995, pp. 56–66
10. T.D. Han, T.S. Abdelrahman, Reducing branch divergence in GPU programs, in *Proceedings of the Fourth Workshop on General Purpose Processing on Graphics Processing Units*, 2011, pp. 3:1–3:8
11. S. Carrillo, J. Siegel, X. Li, A control-structure splitting optimization for gpgpu, in *Proceedings of the 6th ACM Conference on Computing Frontiers*, 2009, pp. 147–150
12. W. Fung, I. Sham, G. Yuan, T. Aamodt, Dynamic warp formation and scheduling for efficient GPU control flow, in *40th Annual IEEE/ACM International Symposium on Microarchitecture, 2007. MICRO 2007*, Dec 2007, pp. 407–420
13. S. Sarkar, S. Mitra, A profile guided approach to optimize branch divergence while transforming applications for GPUs, 02 2015, pp. 176–185
14. J.C.H. Park, M. Schlansker, On predicated execution, 1991
15. N. Reissmann, T.L. Falch, B.A. Bjørnseth, H. Bahmann, J. Christian Meyer, M. Jahre, Efficient control flow restructuring for GPUs, in *2016 International Conference on High Performance Computing Simulation (HPCS)*, 2016, pp. 48–57
16. T.G. Rogers, M. O’Connor, T.M. Aamodt, Divergence-aware warp scheduling, in *2013 46th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2013, pp. 99–110
17. L. Yu, X. Tang, M. Wu, T. Chen, Improving branch divergence performance on gpgpu with a new pdom stack and multi-level warp scheduling. *J. Syst. Archit.* **60**, 01 (2013)
18. M.L. Anido, A. Paar, N. Bagherzadeh, Improving the operation autonomy of SIMD processing elements by using guarded instructions and pseudo branches, in *DSD ’02: Proceedings of the Euromicro Symposium on Digital Systems Design*, vol. 0 (IEEE Computer Society, 2002), pp. 148–155
19. I. Chakroun, M. Mezmaiz, N. Melab, A. Bendjoudi, Reducing thread divergence in a GPU-accelerated branch-and-bound algorithm. *Concurr. Comput. Pract. Exper.* **25**(8), 1121–1136 (2013)

20. L.J. Vespa, N. Weng, Gpep: Graphics processing enhanced pattern-matching for high-performance deep packet inspection, in *Proceedings of the 2011 International Conference on Internet of Things*, Washington, DC, 2011, pp. 74–81
21. Y. Liang, M.T. Satria, K. Rupnow, D. Chen, An accurate GPU performance model for effective control flow divergence optimization. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **35**, 1165–1178 (2016)
22. Z. Yu, L. Eeckhout, C. Xu, Thread similarity matrix: Visualizing branch divergence in gpgpu programs, in *2016 45th International Conference on Parallel Processing (ICPP)*, 2016, pp. 179–184
23. H. Lin, C.-L. Wang, H. Liu, On-GPU thread-data remapping for branch divergence reduction. *ACM Trans. Archit. Code Optim.* **15**(3) (2018) [Online]. Available: <https://doi.org/10.1145/3242089>
24. C. Schied, J. Hanika, H. Dammertz, H. Lensch, Chapter 18 - high-performance iterated function systems, in *GPU Computing Gems Emerald Edition*, ed. by W. mei W. Hwu (Morgan Kaufmann, Boston, 2011), pp. 263–273
25. M. Huang, W. Yang, Partial flattening: A compilation technique for irregular nested parallelism on gpgpus, in *2016 45th International Conference on Parallel Processing (ICPP)*, 2016, pp. 552–561
26. B. Grigorian, G. Reinman, Accelerating divergent applications on simd architectures using neural networks. *ACM Trans. Archit. Code Optim.* **12**(1), 1–23 (2015)
27. L. Vespa, M. Mathew, N. Weng, P3fsm: Portable predictive pattern matching finite state machine, in *20th IEEE International Conference on Application-specific Systems, Architectures and Processors*, Boston, 2009, pp. 219–222

Prototype of MANET Network with Ring Topology for Mobile Devices



Ramses Fuentes Pérez, Erika Hernández Rubio, Diego D. Flores Nogueira,
and Amilcar Meneses Viveros

1 Introduction

There are different scenarios where it is difficult to have an Internet connection and mobile device users require to share information with each other. Collaborative applications such as games, photo-trapping, or Internet of Things systems are examples where connectivity between users is necessary, regardless of whether or not there is an Internet connection [1, 2, 5]. *Mobile ad hoc networks* (MANET) solve this problem by creating dynamic networks from the mobile nodes that want to participate in it [4].

The MANET networks are autonomous and self-organized without a fixed topology. This kind of networks can work everywhere and does not require a static access point. Each node of a MANET network functions as a router and host at the same time, it allows the communication between nodes that are not on its range through middle nodes, these nodes can get in and get out of the network in every moment, and the network will reorganize to continue operating. The fundamental characteristics of a MANET network are distributed operation, multi-hop routing, dynamic topology, shared physical medium, and autonomous and light nodes [1, 3, 4].

R. F. Pérez · D. D. F. Nogueira
ESCOM, Instituto Politécnico Nacional, Mexico City, México

E. H. Rubio
SEPI-ESCOM, Instituto Politécnico Nacional, Mexico City, México
e-mail: ehernandezru@ipn.mx

A. M. Viveros (✉)
Departamento de Computación, Cinvestav-IPN, Mexico City, México
e-mail: ameneses@cs.cinvestav.mx

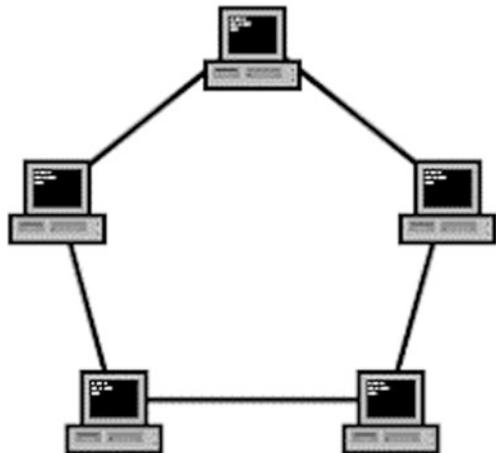
The objective of this work is to get connectivity between heterogeneous devices with different architectures in a transparent and reliable way. For this purpose, it was built a MANET network with ring topology prototype, programming the network access, validation, recovery, restructuring, and message transmission algorithms.

2 MANET Network with Ring Topology Design and Development

The MANET networks require a routing protocol to be able to distribute the messages through the net in a confident way. The routing protocols in MANET networks can be categorized in three types: routing based on the topology, routing based on the localization, and routing based on the energetic consume. The routing networks based on topology require a routing table containing neighbor nodes; in this category we can found the networks with ring topology (see Fig. 1). The advantages of this topology are the fast implementation and the low complexity of the algorithms; the disadvantages are the limited amount of devices and the longer times of propagation, this because to deliver a message it has to pass through all the nodes between the source and destination. The number of nodes in this type of network is limited, depending on the velocity of the nodes.

In this prototype we use IP and MAC address in order to test the basic algorithms in the ring network.

Fig. 1 Ring network topology diagram



2.1 Routing Table

To implement the routing table, a circular double-linked list was used, where each element of the list represents a node from the net and it contains the IP and MAC address, the connection port, and if it is the master node. The first element in the list is the master node. Every node has two neighbors: in the list are the next and previous elements. The identifier for a node is its port and IP address. It was settled an application header for the organization of the network, which contains the identifier of the destination node of a message, the connect, ping, and network flags (see Table 1).

2.2 Network Creation

When starting the program and providing the IP direction and port of the node, the program starts as a master and generates the routing table, then this node is marked as a master in the routing table, and finally the network validation and node discovery processes start.

2.3 Network Login

When starting a new node, provide the new node and another node IP and port; the new node starts as a client, the new node sends a connection message to the provided node (see Table 2), then it gets propagated until it reaches the master, and then the master updates the routing table with the new node and propagates it (see Table 3).

Table 1 Application header
desthdr

desthdr	4 bits	4 bits	4 bits	4 bits
16 bits	port			
16 bits	connect	ping	network	
16 bits	ip			
16 bits				

Table 2 Login message
header

desthdr	4 bits	4 bits	4 bits	4 bits
16 bits	port = new node port			
16 bits	connect = 1	ping = 0	network = 0	
16 bits	ip = new node ip			
16 bits				

2.4 Network Verify

The master node starts a temporizer to periodically verify the network; when the temporizer runs out, it sends a ping to the next node and starts another temporizer waiting a confirmation from the next node (see Table 4); when a node receives a ping, it sends a confirmation to the previous node and a ping to the next until the ping returns to the master – in that moment the network is validated; when a node misses the ping confirmation, it sends a notification to the previous node with the broken node information (see Table 5), and then it gets propagated until it reaches the master; and finally the master restructures the network and propagates the new routing table (see Table 3).

2.5 Message Transmission

The node sends a message with the destination node identifier to the next node; the message propagates until it reaches the destination node (see Table 6).

2.6 Algorithm

The algorithm to handle a message is shown in Fig. 2

Table 3 Updated routing table message header

desthdr	4 bits	4 bits	4 bits	4 bits
16 bits	port = 0			
16 bits	connect = 1	ping = 0	network = network size	
16 bits	ip = 0			
16 bits				

Table 4 Ping message header

desthdr	4 bits	4 bits	4 bits	4 bits
16 bits	port = 0			
16 bits	connect = 0	ping = 1	network = 0	
16 bits	ip = 0			
16 bits				

Table 5 Broken node message header

desthdr	4 bits	4 bits	4 bits	4 bits
16 bits	port = broken node port			
16 bits	connect = 1	ping = 1	network = 0	
16 bits	ip = broken node ip			
16 bits				

Table 7 System test specifications

	Virtual machine	Raspberry PI B+	Jetson TX1
Operative system	Lubuntu	Raspbian	Ubuntu
Processor	Two cores in virtual box	ARM11 with ARMv61 architecture	Quad-Core ARM® Cortex®-A57 MPCore
Architecture	64 bits	32 bits	64 bits
Memory	4GB	512MB	4GB

devices, the algorithm fails because each node is the next and previous of the other, so another logical control was added to handle this case. When connecting a large number of nodes, the time to propagate a message raises linearly, with 16 nodes the mean of the propagation time is 11.2 ms, with 24 nodes 18.5 ms, and with 64 nodes 105 ms. This test interesting since the ring topology have problems have problems with scaling. This affects communication speed between the nodes.

4 Results and Conclusions

The built MANET network allows the transparent connection between heterogeneous devices, with the possibility to get in and get out from the network without affecting the functionality of it; nevertheless, the ring topology was simulated over a network with an access point. The implementation of a MANET network with ring topology is easy, the routing table consists in five fields where two of them are pointers to generate the node list, and the programmed algorithms are simple as they took the advantage of the relation between the ring and the double-linked circular lists, having less than ten steps on each algorithm. The message propagation time with 64 nodes is 105ms which is an acceptable time for applications than don't need a very high connection speed. This prototype has been tested with heterogeneous nodes such as Raspberry PI, Jetson TX1, and laptops among others. As a later work, the implementation of a security protocol in the communications can be done to guarantee the confidentiality and/or the anonymity.

This prototype was originally tested using an access point that assigns IP addresses. The idea in this phase is to test connectivity and error handling on the MANET network. In a next phase, the intention is to use the Bluetooth ports and change the IP addresses to Bluetooth DBADDR addresses and to implement a logical clock.

Acknowledgments The authors thank the facilities and financial support given by the Instituto Politécnico Nacional (SIP project 20201079), as well as the Section of Research and Graduate Studies (SEPI) of ESCOM-IPN and Cinvestav-IPN, provided to accomplish this publication.

References

1. R. Bruzgiene, L. Narbutaite, T. Adomkus, Manet network in internet of things system. *Ad Hoc Netw.* **26**, 89–114 (2017)
2. E. Gromova, D. Gromov, N. Timonin, A. Kirpichnikova, S. Blakeway, A dynamic game of mobile agent placement in a manet, in *2016 International Conference on Systems Informatics, Modelling and Simulation (SIMS)* (IEEE, 2016), pp. 153–158
3. X. Liu, Z. Li, P. Yang, Y. Dong, Information-centric mobile ad hoc networks and content routing: a survey. *Ad Hoc Netw.* **58**, 255–268 (2017)
4. N. Raza, M.U. Aftab, M.Q. Akbar, O. Ashraf, M. Irfan, Mobile ad-hoc networks applications and its challenges. *Commun. Netw.* **8**(3), 131–136 (2016)
5. A.G. Villa, A. Salazar, F. Vargas, Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks. *Eco. Inform.* **41**, 24–32 (2017)

Part VII
International Workshop

New State-of-the-Art Results on ESA's Messenger Space Mission Benchmark



Martin Schlueter, Mohamed Wahib, and Masaharu Munetomo

1 Introduction

The optimization of interplanetary space trajectories is a long-standing challenge for space engineers and applied mathematicians alike. The European Space Agency (ESA) created a publicly available comprehensive benchmark database of global trajectory optimization problems, known as GTOP, corresponding to real-world missions like Cassini, Rosetta, and Messenger. The Messenger (full mission) benchmark in the GTOP database is notably the most difficult instance among those set, resembling an accurate model of the entire trajectory of the original Messenger mission launched by NASA in 2004.

The GTOP database expresses each benchmark as optimization problem (1) with box constraints, whereas the objective function $f(x)$ is considered as nonlinear black-box function depending on a n -dimensional real valued vector of decision variables x . The GTOP database addresses researchers to test and compare their optimization algorithms on the benchmark problems.

$$\text{Minimize } f(x) \quad (x \in \mathbb{R}^n) \tag{1}$$

$$\text{subject to } : x_l \leq x \leq x_u \quad (x_l, x_u \in \mathbb{R}^n)$$

M. Schlueter (✉) · M. Munetomo
Information Initiative Center, Hokkaido University, Sapporo, Japan
e-mail: schlueter@midaco-solver.com; info@midaco-solver.com; munetomo@iic.hokudai.ac.jp

M. Wahib
AIST-Tokyo Tech Real World Big-Data Computation Open Innovation Laboratory, National Institute of Advanced Industrial Science and Technology, Tokyo, Japan
e-mail: mohamed.attia@aist.go.jp

The benchmark instances of the GTOPI database are known to be very difficult to solve and have attracted a considerable amount of attention in the past. Many researchers have worked and published results on the GTOPI database, for example, [1, 3, 5, 6, 8, 10, 12, 13, 15, 17–19, 26], or [28]. A special feature of the GTOPI database is that the actual global optimal solutions are in fact unknown, and thus the ESA/ACT accepts and publishes a new solution that is at least 0.1% better (relative to the objective function value) than the current best known solution. Table 1 lists the individual GTOPI benchmark instances together with their number of solution submissions and the total time span between the first and last submission, measured in years. Note that as of 2020 the original GTOPI database is no longer actively maintained by ESA. However an extended version, named GTOPIX, continues the original GTOPI source code base and introduces some minor improvements and simplified user-handling for the programming languages C/C++, Python, and Matlab. The GTOPIX software is freely available for downloaded at [11].

From Table 1 it can be seen that the Messenger (full mission) benchmark [9, 11] stands out as being by far the most difficult instance to solve. In most cases it took the community several months to about a year to obtain the putative global optimal solution; however the Messenger (full mission) benchmark is an exception in this regard and required a significant amount of submitted solutions and time span between its first and last submission. Well *over 5 years* were required by the community to achieve the current best known solution to the Messenger (full mission) benchmark. This is a remarkable amount of time and reflects the difficulty of this benchmark, about which the ESA stated on their website [9]:

before the remarkable results...were found, **it was hardly believable** that a computer...could design a good trajectory in complete autonomy without making use of additional problem knowledge. ESA/ACT-GTOPI website, 2020 [9]

This contribution addresses exclusively the Messenger (full mission) benchmark and demonstrates that it is possible to robustly solve this instance close to its putative global optimal solution within 1 h on the Hokkaido University HUCC Grand Chariot computer cluster [14], using 1000 cores for distributed computing. The considered optimization algorithm is called MXHPC, which stands for *MIDACO Extension for High-Performance Computing*. The MXHPC algorithm is a (massive) parallelization framework which executes and operates several instances of the

Table 1 GTOPI database benchmark problems

GTOPI benchmark	Number of submissions	Time between first and last submission
Cassini1	3	0.5 years
GTOC1	2	1.1 years
Messenger (reduced)	3	0.9 years
Messenger (full)	10	5.7 years
Cassini2	7	1.2 years
Rosetta	7	0.5 years
Sagas	1	(one submission)

MIDACO algorithm in parallel and has been especially developed for large-scale computer clusters. The here presented results are a significant improvement over the previous state-of-the-art results published in 2017 [24], where it took 12 h of computing time to achieve similar results on a cluster of comparable computing power.

This paper is structured as follows: The second section introduces the Messenger (full mission) benchmark and highlights its difficulty by referring to some previously published numerical results. The third section describes the MXHPC algorithm in detail. The fourth section presents the numerical results obtained by MXHPC solving the Messenger (full mission) benchmark on a computer cluster. Finally some conclusions are drawn.

2 The Messenger (Full Mission) Benchmark

The Messenger (full mission) benchmark [9, 11] models a multi-gravity assist interplanetary space mission from Earth to Mercury, including three resonant fly-bys at Mercury. The sequence of fly-by planets for this mission is given by Earth-Venus-Venus-Mercury-Mercury-Mercury-Mercury.

The objective of this benchmark is to minimize the total ΔV (change in velocity) accumulated during the full mission, which can be interpreted as reducing the fuel consumption. The benchmark invokes 26 continuous decision variables which are described as follows (Table 2).

The best known solution to the problem was obtained in 2014 by the MXHPC/MIDACO optimization software [22] and holds an objective function value of $f(x) = 1.959$.¹

Table 2 Optimization variables for Messenger benchmark

Variable	Description
1	Launch day measured from 1 Jan 2000
2	Initial excess hyperbolic speed (km/sec)
3	Component of excess hyperbolic speed
4	Component of excess hyperbolic speed
5 ~ 10	Time interval between events
11 ~ 16	Fraction of the time interval after DSM ^a
17 ~ 21	Radius of flyby (in planet radii)
22 ~ 26	Angle measured in planet B plane

^aDSM stands for *Deep Space Manoeuvre*

¹Mingcheng Zuo (China Uni. of Geoscience) was able to refine this solution, so it rounds to an objective function value of $f(x) = 1.958$.

2.1 Published Results on Messenger (Full Mission)

While being publicly available for over 10 years now, published numerical results on the Messenger (full mission) benchmark remain very few only. This fact seems to stem from the tremendous difficulty to solve this problem instance. To our best knowledge, Table 4 lists all current available publications reporting numerical results on Messenger (full mission) in chronological order (Table 3).

Table 4 lists the publication date, authors, reference, and name of the considered algorithm together with the overall best objective function value $f(x)$ obtained by that algorithm within that particular study. From Table 4 it can be seen that published results significantly vary and only the 2017 publication achieved a value close to the best known solution of 1.959. It is to note that the 2017 study (Schlueter et al. [24]) and 2019 study (Shunka [25]) both applied massive parallelization to execute their algorithms. The drastic difference in the best achieved solutions between those

Table 3 Best known solution for Messenger (full mission)

Variable	Lower bound	Solution value	Upper bound	Unit
1	1900	2037.8595972244	2300	MJD2000
2	2.5	4.0500001697	4.05	km/sec
3	0	0.5567269199	1	n/a
4	0	0.6347532625	1	n/a
5	100	451.6575153013	500	days
6	100	224.6939374104	500	days
7	100	221.4390510408	500	days
8	100	266.0693628875	500	days
9	100	357.9584322778	500	days
10	100	534.1038782374	600	days
11	0.01	0.6378086222	0.99	days
12	0.01	0.7293472066	0.99	n/a
13	0.01	0.6981836705	0.99	n/a
14	0.01	0.7407197230	0.99	n/a
15	0.01	0.8289833176	0.99	n/a
16	0.01	0.9028496299	0.99	n/a
17	1.1	1.8337484775	6	n/a
18	1.1	1.1000000238	6	n/a
19	1.05	1.0499999523	6	n/a
20	1.05	1.0499999523	6	n/a
21	1.05	1.0499999523	6	n/a
22	$-\pi$	2.7481808788	π	n/a
23	$-\pi$	1.5952416573	π	n/a
24	$-\pi$	2.6241779073	π	n/a
25	$-\pi$	1.6276418577	π	n/a
26	$-\pi$	1.6058416537	π	n/a

Table 4 Published results on Messenger (full mission)

Date	Author(s)	Ref	Algorithm	Best $f(x)$
2010	Biscani et. al.	[6]	PAGMO	3.950
2011	Stracquadanio et. al.	[26]	SADE	2.970
2011	Bryan	[7]	IGATO	7.648
2014	Schlueter	[23]	MIDACO	3.774
2017	Schlueter et al.	[24]	MXHPC	1.961
2019	Shuka	[25]	PASS	8.357

two studies (1.961 vs 8.357) illustrates that the use of *a super-computer alone is not sufficient* to solve the Messenger (full mission) benchmark and that instead the algorithmic element is crucial.

3 The MXHPC/MIDACO Algorithm

The here considered algorithm is called MXHPC and was originally introduced in 2017 (see [24]). MXHPC is a parallelization framework built on top of MIDACO, which is an evolutionary black-box solver (see [21]). As this framework is particularly suited for massive parallelization used in *high-performance computing* (HPC), it is called MXHPC, which stands for *MIDACO Extension for HPC*. The purpose of the MXHPC algorithm is to execute several instances of MIDACO in parallel and manage the exchange of best known solution among those MIDACO instances. The here presented version of MXHPC differs from the original proposed one by a dynamic instead of a static exchange rule, which is illustrated in Sect. 3.1.

Figure 1 illustrates how the MXHPC algorithm executes a number of S different instances of MIDACO in parallel. In regard to the well-known master/slave concept in distributed computing, the individual MIDACO instances can be referred to as slaves, while the MXHPC algorithm can be referred to as master. In evolutionary algorithms such approach is also denoted as coarse-grained parallelization. Note in Fig. 1 that the best known solution is exchanged by MXHPC between individual MIDACO instances at a certain frequency (measured in function evaluation).

The MXHPC algorithm implies several individual parameters; this is the number of MIDACO instances, the exchange frequency of current best known solution, and the survival rate of individual MIDACO instances at exchange times.

Parameter	Description
S	Number of MIDACO instances (called <i>slaves</i>)
<i>exchange</i>	Solution exchange frequency among slaves
<i>survive</i>	Survival rate (in percentage) among slaves

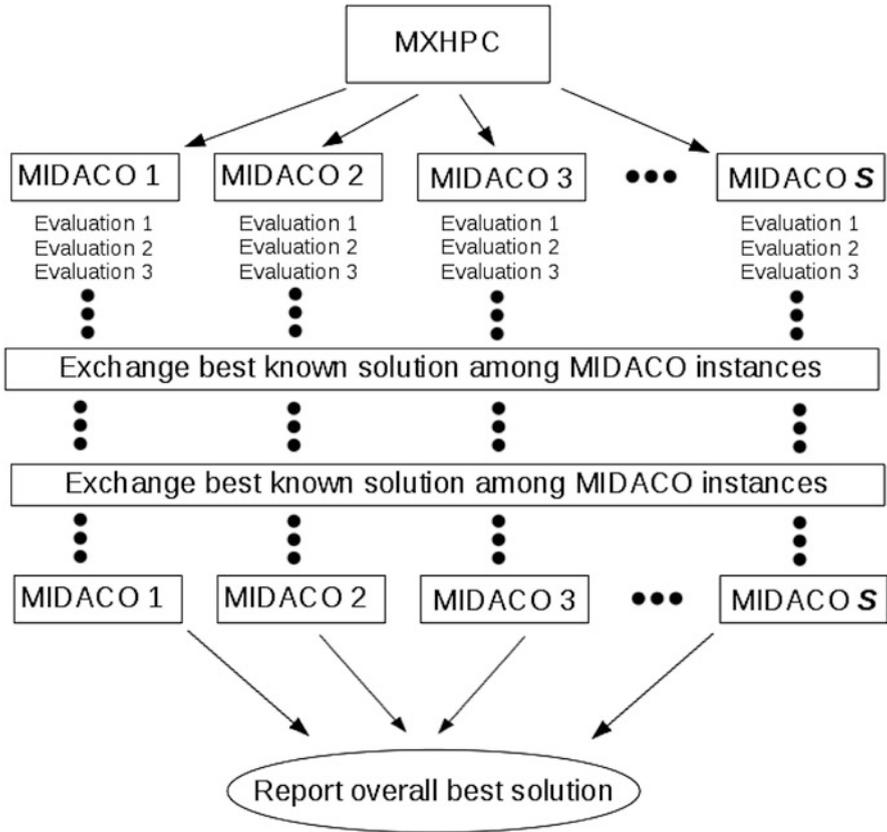


Fig. 1 Illustration of the MXHPC, executing S instances of MIDACO in parallel

The considered exchange mechanism of best known solutions among individual MIDACO instances should be explained in more detail now, as this algorithmic step resembles the most sensitive part of the MXHPC algorithm. Let *survive* be the percentage (e.g., 25%) of surviving MIDACO instances at some *exchange* (e.g., 1,000,000 function evaluation) time of the MXHPC algorithm. Then, at an exchange time, MXHPC will first collect the current best solutions of each of the S individual MIDACO instances and identifies the *survive* (e.g., 25%) best among them. Those MIDACO instances, which hold one of those best solutions, will be unchanged (thus the instance “survives” the exchange procedure). All other MIDACO will be restarted using the overall best known solution as starting point. Readers with a deeper interest in the algorithmic details of MIDACO are referred to [20].

3.1 New Modification: Dynamic Exchange

In contrast to the static exchange rule used within MXHPC in the previous publication from 2017 [24], a dynamic exchange rule is considered in this study. Based on some initial value (called *basevalue*), the evaluation budget of each individual MIDACO run within the MXHPC framework (see Fig. 1) is linearly increased successively. The pseudocode in Algorithm 1 describes in detail how the evaluation budget of each individual MIDACO instance is calculated, according to the successive number of exchanges. Note that a base value of 100000 was used for the numerical tests presented in Sect. 4.

Algorithm 1 Dynamic Exchange (pseudo code)

```

set basevalue = 100000
initialize evaluationbudget = 0

for exchange = 1 : ∞
    evaluationbudget = evaluationbudget +
        exchange × basevalue
end

```

This dynamic exchange rule is based on the idea that with further progress each individual MIDACO instance within MXHPC requires more time (aka more function evaluation) to achieve progress, while such large budgets are not as useful in the beginning of the MXHPC execution. According to the pseudocode in Algorithm 1, the individual evaluation budgets of MIDACO will look as the following:

```

eval-budget for MIDACO until 1st exchange: 100000
eval-budget for MIDACO until 2nd exchange: 300000
eval-budget for MIDACO until 3rd exchange: 600000
and so on ...

```

4 Numerical Results of MXHPC on the Messenger (Full Mission) Benchmark

This section presents the numerical results obtained by MXHPC on the Messenger (full mission) benchmark. All results were calculated on the Hokudai supercomputer (HUCC Grand Chariot [14]) utilizing 1000 cores for distributed computing, which are composed of Intel Xeon Gold 6148 CPUs with a clock rate of 2.7 GHz. Ten

independent test runs of MXHPC have been applied, each using the original lower bounds (see Table 3) as starting point and a different random seed for MIDACO's internal pseudo-random number generator. Each individual test run was allowed to execute for 1 h and then stopped automatically. The following parameters have been used for the MXHPC algorithm.

Parameter	Description
S	1000
<i>exchange</i> (basevalue)	100,000
<i>survive</i>	25%

Table 5 reports the characteristics of each individual test run and the averaged values out of ten runs. The number of total function evaluation is displayed as multitude of 1000 in Table 5, corresponding to the 1000 cores which were used by MXHPC for the parallelization framework.

From Table 5 it can be seen that each run converged to an objective function value roughly above $f(x) = 2.0$ within 1 h of run time. The averaged converged objective function value is $f(x) = 2.0206$ corresponding to the enormous number of about 44×10^9 function evaluation in total. In addition to Table 5, Fig. 2 illustrates the convergence curves in semi-log scale of all ten test runs. Note in Fig. 2 that all test runs have converged below an objective function value of $f(x) = 6.0$ within 1000 s (~ 15 min) of CPU run time.

4.1 Comparison with Previous Results from 2017

This subsection gives an in-depth comparison between the newly presented results and the previously published ones in 2017 (see [24]). Table 6 lists the ten from

Table 5 Ten test runs of MXHPC on Messenger

Run	f(x)	Evaluation	Time [sec]
1	2.0208	41,500,000 \times 1000	3626.35
2	2.0245	41,500,000 \times 1000	3637.20
3	2.0142	43,500,000 \times 1000	3654.09
4	2.0242	48,500,000 \times 1000	3639.93
5	2.0187	43,500,000 \times 1000	3654.51
6	2.0182	45,500,000 \times 1000	3602.26
7	2.0225	44,500,000 \times 1000	3642.45
8	2.0264	47,500,000 \times 1000	3667.15
9	2.0143	43,500,000 \times 1000	3679.37
10	2.0220	42,500,000 \times 1000	3647.33
Average:	2.0206	44,200,001 \times 1000	3645.06

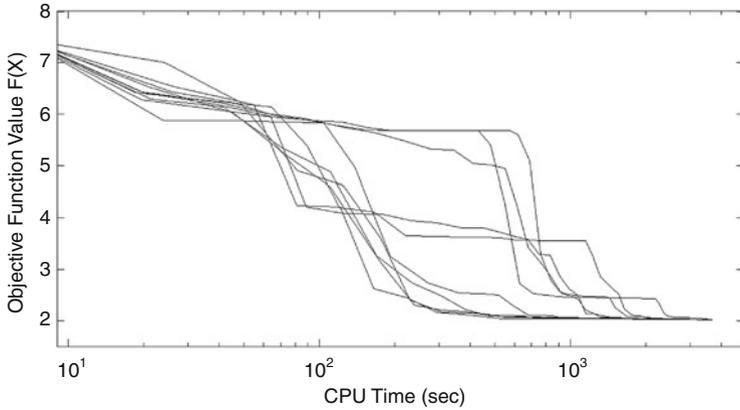


Fig. 2 Convergence curves of ten individual MXHPC test runs

Table 6 Comparison regarding overall solution quality

Run	New results (2020)		Previous results (2017)	
	f(x)	Evaluation	f(x)	Evaluation
1	2.0208	41,500 × 10 ⁶	2.0225	70,000 × 10 ⁶
2	2.0245	41,500 × 10 ⁶	2.0295	69,000 × 10 ⁶
3	2.0142	43,500 × 10 ⁶	2.0313	72,000 × 10 ⁶
4	2.0242	48,500 × 10 ⁶	2.0481	71,000 × 10 ⁶
5	2.0187	43,500 × 10 ⁶	2.0449	67,000 × 10 ⁶
6	2.0182	45,500 × 10 ⁶	2.0481	47,000 × 10 ⁶
7	2.0225	44,500 × 10 ⁶	2.0379	67,000 × 10 ⁶
8	2.0264	47,500 × 10 ⁶	2.0441	69,000 × 10 ⁶
9	2.0143	43,500 × 10 ⁶	2.0528	71,000 × 10 ⁶
10	2.0220	42,500 × 10 ⁶	2.0263	68,000 × 10 ⁶
∅	2.0206	44,200 × 10 ⁶	2.0386	67,100 × 10 ⁶

scratch² test runs of MXHPC with their corresponding final objective function value f(x) and corresponding number of total function evaluation. Both results, those from 2017 and 2020, have been calculated on a cluster with 1000 cores. Back in 2017, this was a Fujitsu FX10 cluster [2], while now in 2020 this was the HUCC Grand Chariot cluster of the Hokkaido University [14].

From Table 6 it can be seen that both series of results averaged at a similar final solution objective function value of 2.0206 and 2.0386. The percentual difference between those values is about 0.9%, which may appear small but is in fact relevant in context of the difficulty of the Messenger benchmark. In regard to the number

²From scratch means here that the lower bounds have been used for all test runs. Those test runs therefore aim at exploring the entire search space and are not refinements of previous found solutions.

of total function valuation, the new results require roughly about half (58.85%) of the amount required in 2017. As the results from 2017 were calculated with a time limit of 12 h for each run while the new results are required in only 1 h for each run, the averaged number of function evaluation reveals that the HUCC Grand Chariot cluster performed around five times ($4.938 = 12 \times (1 - 0.5885)$) faster than the Fujitsu FX10 cluster.

Given that the new results required only around half the amount of function evaluation as back in 2017 and that the averaged final solution objective function value still shows an improvement of 0.9% over the old results, it can be concluded that the dynamic exchange strategy (see Sect. 3.1) is improving the algorithmic performance about at least two times.

An interesting threshold value for the Messenger (full mission) benchmark is the objective function value of $f(x) = 2.113$. The solution corresponding to this value was obtained by G. Stracquadiano (Johns Hopkins University) and G. Nicosia (University of Catania) and was included as new record solution in the GTOP database on 10 April 2012. Such solution is an improvement over the previous published one by Stracquadiano et al. [26], and to this date it remains the best known solution that was found without utilizing the MIDACO algorithm. It is therefore the best known competitive result and can act as a reference to compare with. In [24] the CPU time to reach the threshold of 2.113 has been measured and reported. Table 7 reports for each of the ten current MXHPC test runs after what amount of function evaluation and CPU time and with what objective function value the threshold of 2.113 was breached. Additionally Table 7 included the CPU time to breach the threshold in the 2017 study [24].

From Table 7 it can be seen that back in 2017 it took on average 10,263.5 s to breach the threshold value of 2.113, while it took 882.4 s on average in this study. That is a 11.6 times improvement in such regard. Note that in Table 7 the number of function evaluation required to breach the threshold is on average about 15×10^9 , while the number of function evaluation performed in a full hour in Table 5

Table 7 Comparison regarding breach of $f(x) = 2.113$

Run	New results (2020)			Results (2017)
	$f(x)$	Evaluation	Time [sec]	Time [sec]
1	2.112	$5,500 \times 10^6$	467	9,430
2	2.091	$5,500 \times 10^6$	473	7,261
3	2.107	$8,500 \times 10^6$	678	13,284
4	2.084	$25,500 \times 10^6$	1,593	12,344
5	2.098	$17,500 \times 10^6$	1,331	5,170
6	2.068	$18,500 \times 10^6$	1,312	14,104
7	2.091	$23,500 \times 10^6$	1,801	3,563
8	2.090	$33,500 \times 10^6$	242	7,794
9	2.080	$5,500 \times 10^6$	450	23,273
10	2.101	$5,500 \times 10^6$	477	6,412
∅	2.092	$14,900 \times 10^6$	882.4	10,263.5

averages about 44×10^9 . This means that about one third of the total evaluation budget (aka CPU time) is spent by MXHPC on reaching a value of about 2.092 and then two thirds of the budget is spent by MXHPC on further converging toward a value of about 2.0206. This observation indicates that the Messenger (full mission) benchmark is exceptionally difficult in both regards: locating the global optimal valley and then converging into that area to the exact solution.

5 Conclusions

Over 10 years, ESA's Messenger benchmark is publicly available and acts as one of world's most challenging real-world benchmarks, formulated as numerical black-box optimization problem. In 2017, it could be shown for the very first time that it is possible to solve this benchmark in a fully automatic way by applying an evolutionary algorithm on a supercomputer. While few publications attempted to address the Messenger problem, none were able to solve it to a close optimal solution, except for the 2017 study (see Table 4). This is in particular true for the suboptimal results published by Shuka [25], which also applied various evolutionary strategies on a supercomputer.

The results presented in this contribution are a continuation of the 2017 study and exhibit a significant improvement in terms of CPU run-time and evaluation budget. While in 2017 it took about 12 h, the here presented results require only 1 h of run-time and about half the function evaluation budget to achieve a similar (even slightly better) solution quality (see Table 6). An in-depth analysis in Sect. 4.1 revealed that this performance gain was about five times due to the faster hardware and about two times due to the algorithmic change made within MXHPC described in Sect. 3.1.

While the novelty of the algorithmic contribution in this study is only incremental, the reported numerical results are still of significance to a broad community of researchers who utilize these kinds of benchmarks. This is due to the tremendous difficulty of the Messenger benchmark, about which the ESA stated that it was hardly believable to be solvable in an automatic fashion at all (see quote in Sect. 1). While the 2017 study proved the automatic solubility of this benchmark, this study is able to reduce the required CPU run-time to a single hour to robustly achieve a near-optimal solution. Overall, the here presented results fortify the effectiveness of massively parallelized evolutionary computing for complex real-world problems which have been previously considered intractable.

References

1. B. Addis, A. Cassioli, M. Locatelli, F. Schoen, Global optimization for the design of space trajectories. *Comput. Optim. Appl.* **48**(3), 635–652 (2011)
2. AIST Artificial Intelligence Cloud (AAIC). https://www.airc.aist.go.jp/en/info_details/computer-resources.html (2020)

3. C. Ampatzis, D. Izzo, Machine learning techniques for approximation of objective functions in trajectory optimisation, in *Proceedings of International Conference Artificial Intelligence in Space (IJCAI)* (2009)
4. A. Auger, N. Hansen, A restart CMA evolution strategy with increasing population size, in *IEEE Congress on Evolutionary Computation, Proceedings* (IEEE, 2005), pp. 1769–1776
5. M. Biazzini, B. Banhelyi, A. Montresor, M. Jelasity, Distributed hyper-heuristics for real parameter optimization, in *Proceedings 11th Annual Conference Genetic and Evolutionary Computation (GECCO)*, pp. 1339–1346 (2009)
6. F. Biscani, D. Izzo, C.H. Yam, A global optimisation toolbox for massively parallel engineering optimisation, in *Proceedings 4th International Conference Astrodynamics Tools and Techniques (ICATT)* (2010)
7. J.M. Bryan, Global optimization of MGA-DSM problems using the Interplanetary Gravity Assist Trajectory Optimizer (IGATO), Master Thesis, California Polytechnic State University (USA) (2011)
8. G. Danoy, C. Pinto, B. Dorronsoro, P. Bouvry, New state-of-the-art results for cassini2 global trajectory optimization problem. *Acta Futura* **5**, 65–72 (2012)
9. European Space Agency (ESA) and Advanced Concepts Team (ACT). GTOP database – global optimisation trajectory problems and solutions, archived webpage https://www.esa.int/gsp/ACT/projects/gtop/messenger_full/ (2020)
10. A.H.G.E. Gad, Space trajectories optimization using variable-chromosome-length genetic algorithms. PhD-Thesis, Michigan Technological University (2011)
11. GTOPX – Space Mission Benchmark Collection, software available at <http://www.midaco-solver.com/index.php/about/benchmarks/gtopx> (2020)
12. A. Gruber, Multi Gravity Assist Optimierung mittels Evolutionsstrategien, BSc-Thesis. Vienna University of Technology (2009)
13. T.A. Henderson, A Learning Approach To Sampling Optimization: Applications in Astrodynamics, Ph.D.-Thesis, Texas A & M University (2013)
14. Hokaido University High-Performance Intercloud. <https://www.hucc.hokudai.ac.jp/en/supercomputer/sc-overview/> (2020)
15. S.K.M. Islam, S.G.S. Roy, P.N. Suganthan, An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. *IEEE Trans. Syst. Man Cybern.* **42**(2), 482–500 (2012)
16. D. Izzo, 1st ACT global trajectory optimisation competition: Problem description and summary of the results. *Acta Astronaut.* **61**(9), 731–734 (2007)
17. D. Izzo, Global optimization and space pruning for spacecraft trajectory design, in *Spacecraft Trajectory Optimization* ed. by B. Conway (Cambridge University Press, 2010), pp. 178–199
18. A. Lancinskis, J. Zilinskis, P.M. Ortigosa, Investigation of parallel particle swarm optimization algorithm with reduction of the search area, in *Proceedings of International Conference Cluster Computing Workshops and Posters* (IEEE, 2010)
19. P. Musegaas, Optimization of Space Trajectories Including Multiple Gravity Assists and Deep Space Maneuvers, *MSc Thesis*, Delft University of Technology (2012)
20. M. Schlueter, J.A. Egea, J.R. Banga, Extended ant colony optimization for non-convex mixed integer nonlinear programming. *Comput. Oper Res.* **36**(7), 2217–2229 (2009)
21. M. Schlueter, M. Gerdt, J.J. Rueckmann, A numerical study of MIDACO on 100 MINLP benchmarks. *Optimization* **61**(7), 873–900 (2012)
22. M. Schlueter, S. Erb, M. Gerdt, S. Kemble, J.J. Rueckmann, MIDACO on MINLP space applications. *Adv. Space Res.* **51**(7), 1116–1131 (2013)
23. M. Schlueter, MIDACO software performance on interplanetary trajectory benchmarks. *Adv. Space Res.* **54**(4), 744–754 (2014)
24. M. Schlueter, M. Wahib, M. Munetomo, Numerical optimization of ESA’s Messenger space mission benchmark, in *Proceedings of the Evostar Conference* (Springer, Amsterdam, 2017) Apr 19–21, pp. 725–737
25. R. Shuka, Parallele adaptive Schwarmuche fuer Blackbox-Probleme. Ph.D.-Thesis, Gottfried Wilhelm Leibniz University Hannover (2018)

26. G. Stracquadanio, A. La Ferla, M. De Felice, G. Nicosia, Design of robust space trajectories, in *Proceedings of 31st International Conference Artificial Intelligence (SGAI)* (2011)
27. M. Ceriotti, M. Vasile, MGA trajectory planning with an ACO-inspired algorithm. *Acta Astronaut.* **67**(9–10), 1202–1217 (2010)
28. T. Vinko, D. Izzo, Global Optimisation Heuristics and Test Problems for Preliminary Spacecraft Trajectory Design, European Space Agency, ACT Technical Report. ACT-TNT-MAD-GOHTPPSTD (2008)

Crawling Low Appearance Frequency Character Images for Early-Modern Japanese Printed Character Recognition



Nanami Fujisaki, Yu Ishikawa, Masami Takata, and Kazuki Joe

1 Introduction

Japanese character recognition research consists of handwritten and printed character recognition. The latter was already matured, and OCR is commercially available. The former had reached a recognition rate of more than 99% by the turn of the twenty-first century and was considered to have little room for further research. We found that there is the third Japanese character recognition research that is for early-modern (1968–1945) printed books where typographical printing was used in publications, which is quite different from today's standardized fonts. There are more than 20,000 publishers for early-modern Japanese printed books. Different publishers generate different early-modern printed characters in different publication years. So we proposed the third Japanese character recognition method [1–3] with a small number of learning data taken from National Diet Library Digital Collections [4] where 35,000 titles of early-modern Japanese printed books are open to the public as picture images. In [5], we presented that a CNN is better than our previous recognition methods for early-modern Japanese printed books where we used six sets of data: each set has 2678 types of characters. When the CNN is trained with just collected data sets, the performance is as same as our previous methods. However, we found that the recognition rate for unknown data reaches to more than 97% when 21 sets of the current JIS level 1 (3011 Japanese character types) font sets as well as the five sets of the above 2678 types are used for the training data. The rest one set is used as test data.

N. Fujisaki (✉) · M. Takata · K. Joe
Nara Women's University, Nara, Japan
e-mail: fujisaki-nanami1616@lics.nara-wu.ac.jp

Y. Ishikawa
Shiga University, Shiga, Japan

The breakthrough in [5] is to show that practical early-modern Japanese printed character recognition is possible with a small number of early-modern Japanese printed character sets and enough numbers of current fonts. So far, we have been conducting preliminary experiments of early-modern Japanese printed character recognition using a character set that is not even at the JIS level 1. In order to carry out practical early-modern Japanese printed character recognition, it is necessary to secure a data set that includes the JIS level 2 (3390 Japanese character types). As we discuss later in this paper, in the character collection, the appearance frequency has a stronger effect on the appearance probability than Zipf's law [6]. In the case of low appearance frequency (less than 2000th in JIS levels 1 and 2), the appearance probability is expected to be a few millionth parts, and such collection is extremely difficult. Based on our experience with early-modern Japanese printed character collecting, it is possible to collect characters by hand with an appearance probability of only a few thousand parts, but when the probability is hundreds of thousands, automation is required, and the probability of a few millionth parts is currently impossible to achieve. In [7], we proposed a structure in the population to be collected in order to collect JIS level 2, i.e., character types with an appearance probability of one in a few million. In other words, by dividing the books to be collected into the domains of literature, economics, medicine, science, engineering, etc., we can obtain the distribution of the appearance frequency for each domain and automate the character collection so that a character type with a low appearance frequency in one domain can be easily collected in another domain.

The structure of this paper is as follows. In Sect. 2, the method of collecting learning data for early-modern Japanese printed character recognition by a web application and the appearance frequency of character types in the collecting work are explained. In Sect. 3, we propose a learning data collection method using crawlers. In Sect. 4, we show the operation results of the crawler we developed.

2 Collecting Learning Data for Early-Modern Japanese Printed Character Recognition

2.1 Web Application for Collecting Learning Data

In order to improve the early-modern Japanese printed character recognition, it is necessary for us to collect character images, which are learning data, from as many different years of publication and publishers as possible. Therefore, a web application was proposed to improve the efficiency of the character collection [8]. The web application is designed to meet the demands that are easy to access and operate so that more users can participate in the collection work. We describe the procedure of the collection work using the web application. First, when a book is selected, the book image is preprocessed so that character recognition of the character image extracted from the book image is performed. When the processing

is completed, the original image of the book page and the text obtained from the recognition results are displayed on the web page. The user confirms whether the character recognition is correct. In the case of incorrect recognition, the user modifies the result. Finally, the recognized character is registered in the character database.

Such web application was developed and expected to be used effectively for the collection work. However, when we actually ran the web application, two problems occurred. The one was that the operation of the web application became unstable when each function was executed together. The development of web applications was shared by function. However, each function created by an individual lacked uniformity. Therefore, while each function worked on its own, the web application that combined each function into one did not have practical performance. The other problem was that the character database of the training data was incomplete. Characters were registered without an appropriate definition of the database. In addition, to improve character recognition accuracy, the learning data should be by publisher and year of publication. Therefore, it was necessary to improve the character database so that the character database could be registered by distinguishing the publisher and the date of publication.

In order to solve these problems, we redeveloped the web application. In the redevelopment, a new web application framework has been introduced. This allows us to develop smoothly and to create the stable web application. In addition, we reviewed the specification of the web application to extend and improve its functionality. Figure 1 shows the procedure of the collection work by the newly developed web application. The general procedure is the same as the existing method. The collection work proceeds in the order of book select, image preprocessing, character recognition, recognition result display, correction, and database registration. The

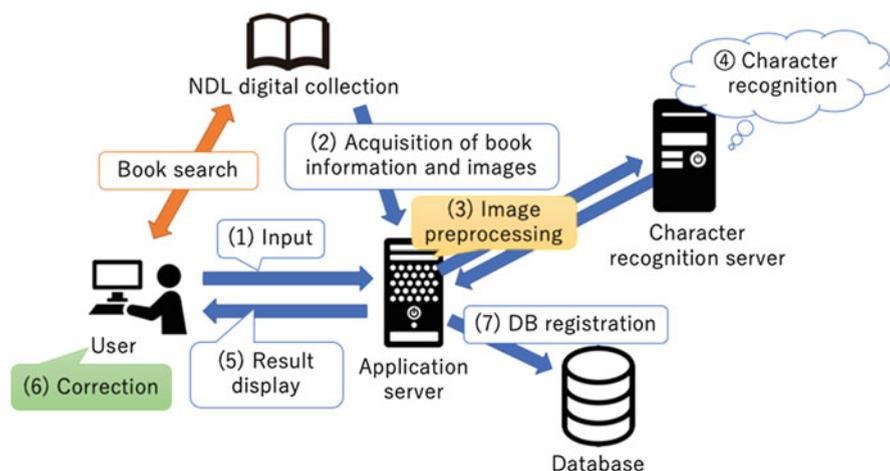


Fig. 1 Procedure of collection work by web application

existing web application requires users to download book images in advance. As a new feature, an automatic download function for book images has been added. After obtaining the book information, the server acquires the book images directly from the National Diet Library Digital Collections by using an API. We also applied a new design to the display of character recognition results. Characters within a low confidence rate of character recognition are highlighted. The left part of Fig. 2 shows the display page of the recognition results. Users are more likely to find potentially misrecognized characters. We also improved the web application so that users can modify the character recognition results on the browser. In the existent web application, to modify the recognition results, the user downloads the recognition result files from the server. Then, after modifying the text, the user had to upload it to the server. In the new web application, the user’s modification work is performed by front-end processing. If the user finds a misrecognized character, he or she clicks on the corresponding text character in the browser. At that time, a window is opened to modify the character. The right part of Fig. 2 shows the actual display of the modification window. The user gives the correct characters in the input form. Corrected characters are reflected in the text on the result display screen in real time. In addition, we modified the table definitions of the character database so that characters are registered using publication year and publisher. The early-modern Japanese printed book table has the following columns: identification number (primary key), book name, publisher, year of publication, and the presence or absence of rubrics. The early-modern Japanese printed character table has columns of JIS code, book number (compound primary key), character image file name, image size, number of black pixels, and binary data of the image.

The redevelopment of the web application is aiming for the stability operation and the user interface. The collection method by this web application is useful for accurately collecting character images from the user-specified books. However, it is difficult to collect a wide variety of learning data just with this method.



Fig. 2 Display of recognition results and character correction window

Characters with a high appearance frequency are to be collected in large numbers, while low appearance frequency characters are extremely difficult to collect. The appearance frequency of characters varies considerably among character types. It is difficult and inefficient to find a character type with low appearance frequency by the conventional method of sequentially selecting books and collecting characters indiscriminately.

2.2 Character Types with Low Appearance Frequency

In this subsection, we investigate the appearance frequency of various character types found in the early-modern Japanese printed books which are also registered in Aozora Bunko [9] in order to confirm the difference of appearance frequency of each character type. Aozora Bunko is a collection of early-modern Japanese books textualized by numerous volunteers. We compare Zipf's law with the distribution of the appearance frequency of characters found in Aozora Bunko. Zipf's law is that when words are sorted in the order of their appearance frequency, the ratio of the n th most frequently appearing character to the total is proportional to $1/n$. Considering that the early-modern Japanese printed books may contain about 6000 types of characters, according to Zipf's law, characters with less than the top 1/3 appearance frequency are extremely difficult to find out. The blue graph in Fig. 3 shows the scatter plot presenting the appearance probability of characters found in Aozora Bunko. In addition, the red and green graph in Fig. 3 represents $1/n$ of Zipf's law and $1/n^2$, respectively, which has lower probability than Zipf's law. The vertical axis shows the logarithm of frequency probability, and the horizontal axis shows the frequency ranking. From Fig. 3, we observe that the blue and red graphs overlap at the top 2000 in appearance frequency. In other words, the appearance frequency of characters in Aozora Bunko satisfies Zipf's law within the top 2000 species. When

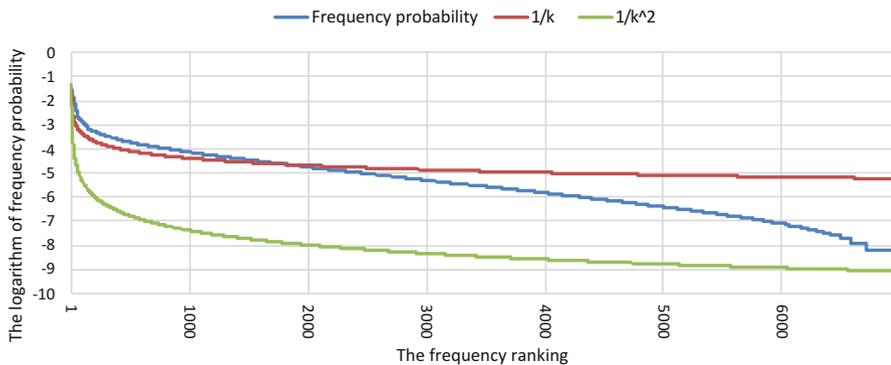


Fig. 3 Comparison of frequency probability of characters in Aozora Bunko with $1/n$ and $1/n^2$

the appearance frequency is lower than 2000, the appearance probability is lower than $1/n$, namely, beyond Zipf's law. The result shows that the number of characters in Aozora Bunko that exceed Zipf's law is about 2000 and the remaining 4000 characters are so difficult to find that they do not even satisfy Zipf's law. Therefore, it is expected that it will be even more difficult to acquire character types with appearance frequency of less than 2000 places.

3 Crawler to Collect Characters with Low Appearance Frequency

In order to improve the accuracy of early-modern Japanese printed character recognition, it is necessary to acquire more character types. However, the conventional method of sequentially collecting characters from books is inefficient in collecting the character types evenly. Therefore, we propose a collection method by crawling as a method to collect efficiently the character types with low appearance frequency that the number of learning data is insufficient. We develop the web application by using the proposed method.

3.1 Collecting Method by Crawler

Figure 4 shows the procedure for collecting characters by a crawler. First, the user accesses the web application and inputs the characters to be searched as well as the year and publisher of the books, which is the range of crawling. When the input data are sent to the server, the server retrieves the book information that satisfies the search conditions through the API. After that, the server downloads the images

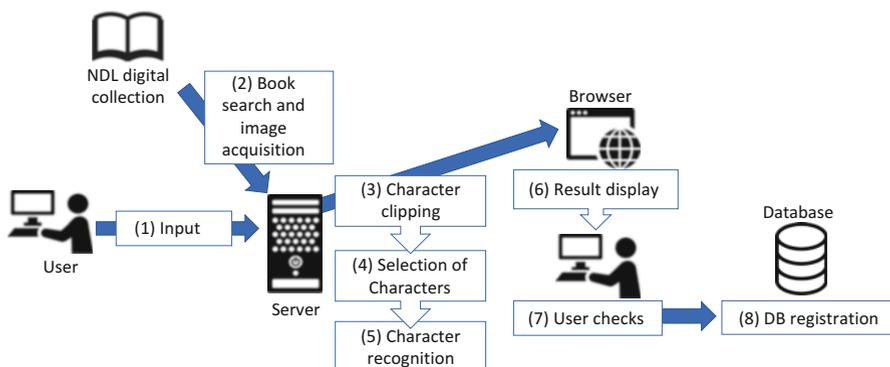


Fig. 4 Procedure for collecting characters by crawler

Fig. 5 Top five characters in high appearance frequency and bottom five characters in low appearance frequency

人	一	見	出	日
齧	逃	麩	黼	穉

of the corresponding books in turn. Next, the server clips the characters from the downloaded book images. The clipped character images are sorted out according to the black pixel ratio before character recognition. Figure 5 shows characters with high and low appearance frequency of early-modern Japanese printed characters found in Aozora Bunko. The upper and the lower row of Fig. 5 shows the top five and the bottom five frequently used Kanji characters, respectively. Characters with high appearance frequency, such as those in the upper part of Fig. 5, tend to consist of a small number of strokes. Therefore, it is considered that there are relatively many characters with a small proportion of black pixels in the character image. On the other hand, as shown in the lower part of Fig. 5, the character types with low appearance frequency have many strokes and are complicated. Therefore, it is expected that the proportion of black pixels is high. By using this tendency, the pixels of each clipped character image are examined, and the character image with a low black pixel rate is discarded. Just the character images that are likely to be with low appearance frequency are left, and they are regarded as the target for character recognition, which consumes expensive computation cost. After the character selection process by black pixel ratio, character recognition is performed. After the character recognition, the JIS code indicating the recognition result is sent back. Then, the user checks the search results against the desired characters. First, the character image whose candidate recognition result is applicable to the desired character is sent to the browser. The browser displays the character image as the search result on the web page for each input character. The user checks the displayed character images and selects the one that is recognized correctly. After the selection is completed, the browser sends the selected character image to the server. Finally, the character image and the corresponding JIS code of the recognition result are registered in the character database.

3.2 Implementation

In this subsection, we describe how to implement each function of the collecting method by crawler. We use python for server-side programs and JavaScript, HTML, and CSS for the client-side programs including the user interface. We also adopt Django [10], a web application framework, and PostgreSQL [11].

First, we explain how to implement the function to acquire book information and images. The SRU interface [12] of the search API of the National Diet Library Search [13] is used to search for books in the specified range. A search query is generated from the publisher and year range specified by the user. When the server sends a search request, an XML code containing metadata is returned. By parsing the XML code, we get the number of search results, the identification number of the book, and the copyright information. Books whose images are available on the NDL Digital Collections and downloadable through the API are those whose copyright information is set to “Open to the Internet.” Therefore, only books whose copyright information is described as “Open to the Internet” are targeted for crawling. We also use the presentation API [14] and image API [15] of the IIF interface [16] when acquiring book images. Through the presentation API, the server sends a request URI generated from the book’s identification number and gets the number of page images of the book. Then, using the image API, the server sends a request specifying the page number and retrieves the image in jpg format.

Layout analysis using semantic segmentation [17] is applied to the character clipping process. The character image clipped from the book image is binarized and normalized to a size of 62×62 pixels.

Next, we describe the implementation for selecting character images based on the black pixel ratio, which represents the number of strokes and their complexity of a given character. All extracted character images are binary images. By counting the number of zero value pixels, we can examine the proportion of black areas. The program judges whether the black pixel ratio of the acquired character image is lower than a certain threshold. If the black pixel ratio is above the threshold, the character image is regarded as the target of character recognition. On the other hand, if it is below the threshold, the character image is abandoned. The threshold used for the determination is obtained from the distribution of the black pixel ratio of the character images collected so far. The black pixel percentage of the character image is all stored in a single CSV file, and the program reads the CSV file to perform statistical processing to calculate the distribution. Furthermore, when a character image of a new character type is obtained, the black pixel ratio is added to the CSV file. As a result, the distribution of the black pixel rate is updated as the collection work progresses, and we can expect better appropriate threshold.

The character recognition process uses the CNN to recognize early-modern printed Japanese characters. When a character image is input, the JIS code of the recognition result candidate and its certainty rate are output. When this rate is above a certain value, it is considered as a candidate for the recognition result. The character image whose result candidate matches the character searched for is sent to the browser as the search result.

We describe the implementation detail for the front-end processing. Data communication between the front end and the back end is operated by Ajax. In order to display the progress of the process on the web page, the browser obtains the number of the book and the page in process as appropriate. We also implement the way that the browser receives the results from the server as each book page is processed, rather than displaying the results of the search for all books in the search range at

once. When candidate character images are found, the search results are displayed in sequence. In the process of selecting a text image by the user, a class indicating that the image has been selected is added to the `img` element of the clicked HTML. The character image to which this class has been added is displayed with a check mark on the character image on the web page to indicate that it has been selected. When sending a batch of images to the server after the selection process, the browser can aggregate and send only the character images to which the class has been assigned.

4 Operation Results

In this section, we demonstrate the usefulness of the web application with crawling by checking the operation behavior and presenting the results. We start the development server that comes with the web application framework Django and test it in the local environment.

We verify the front-end behavior of the application. First, the user accesses the input page (the left part of Fig. 6) and fills in the form with the characters to be searched and the year of publication and publisher for the crawling range. The user can input multiple characters to be searched. When the user clicks the submit button after giving the information, the web page transitions to the page that displays the search results. On this result page, a progress bar showing the progress of the process is displayed. The right part of Fig. 6 shows the progress bar on the results display page. There are two types of progress bars: one shows the number of books being crawled, and the other shows how many pages of the book are being processed. When the back-end process is completed, the result of the search is displayed under the progress bar. Figure 7 shows the display part of the search results on the results display page. Character images are displayed in order as soon as they are found. If the character to be searched is not found, it is displayed as “Not applicable.” Processing on the server and getting the results of the processing are executed in the background. Therefore, the user can select the recognition results even during the crawling process. As soon as the result is displayed, the user checks the displayed character image. If there is a character image of the desired character, the user clicks the character image. As shown in the right part of Fig. 7, a check box indicating that

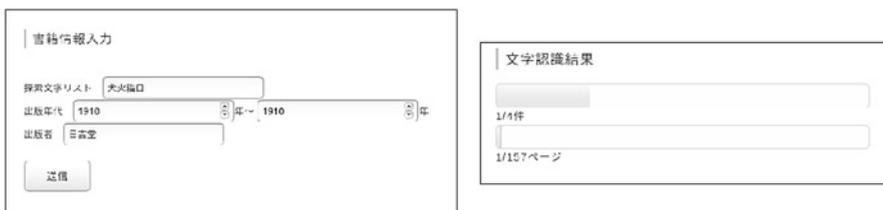


Fig. 6 Input page of the crawler application and the progress bar



Fig. 7 Display of search results and selected character images

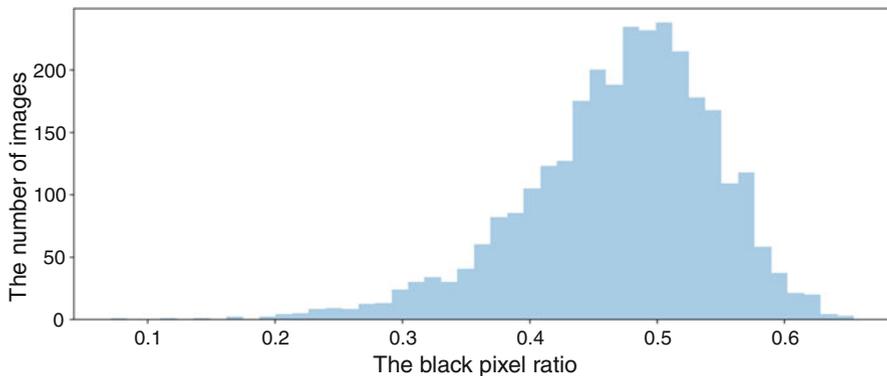


Fig. 8 Distribution of black pixel percentage of character images in the Mincho font

it has been selected appears above the clicked text image. When all the crawling of the specified range of books is completed, the submit button appears on the page that displays the results. When the user finishes the work of selecting a character image that is recognized correctly, he/she clicks the send button. This completes the user's work.

We describe the results of one of the back-end processes, the process of obtaining the black pixel ratio of a character image. For the input data, we use a character image of HG Mincho E, a font used today. The characters are 3011 types of JIS level 1 Hiragana and Kanji. The program acquires the black pixel ratio of each character image and calculates the distribution of the black pixel ratio. Figure 8 shows the histogram of the black pixel ratio of the 3011 character images. The horizontal axis is the black pixel ratio of the character image, and the vertical axis is the number of images. In addition, we investigate the distribution of the black pixel ratio for

each appearance frequency of the characters found in Aozora Bunko mentioned in Sect. 2.2. Figure 9 shows the distribution of the black pixel percentage in HG Mincho E by character appearance frequency. From Fig. 9, we confirm that the characters with the higher frequency of appearance have more characters with lower black pixel ratio. In addition, the distributions range of the black pixel rate gradually moves upward up to the top 1000 characters in the frequency of appearance. On the other hand, the distribution of the black pixel percentage is almost constant for characters with frequency ranks lower than 1000 characters. In other words, the top 1000 characters in the frequency of appearance can be obtained by the black pixel rate. Thereby, the character selection process by black pixel ratio before character recognition is effective. In addition, the distribution of the black pixel percentage of character image is also expected to change depending on font types, since the shape and fineness of the characters are different. Therefore, once a sufficient number of character images have been collected, we calculate the distributions of the black pixel rates by publisher and publication year. It is expected that we can obtain better thresholds.

Finally, we discuss the performance estimates for character collection. The most costly part of character collection is the computation on the character recognition server, where the CNN proposed in [5] consists of three layers of convolutional/pooling layers and a fully connected layer. The first, second, and third convolutional layer uses 160, 320, and 640 7×7 filters, respectively. The input to the first convolutional layer is 64×64 binary character image, while the output of the third convolutional layer is the 8×8 filtering result. In the fully connected layer, [7] is extended to discriminate up to JIS levels 1 and 2. When we run this CNN on GTX1080Ti, it takes about 1.5 seconds to recognize a character image. For our low appearance frequency character collection, let us assume that we were able to recognize 10,000 character images per hour using a high-end GPU such as RTX2080Ti. In this case, a low appearance frequency character of hundreds of

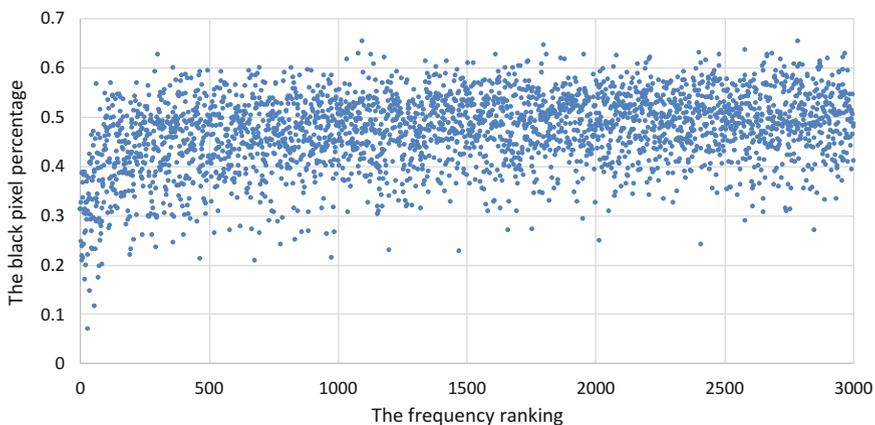


Fig. 9 Distribution of black pixel percentage by the order of appearance frequency of characters

thousands can be detected in about a day. The problem is the case of very low appearance frequency of characters in a few millionth parts, but we can deal with it by scanning many domains equally, as explained in Sect. 1. With this policy, our goal is to have at least ten sets of JIS levels 1 and 2 character types within 1 year.

5 Conclusions

In this paper, we propose a character collection method by a character crawler to acquire characters with low appearance frequency from early-modern Japanese printed books. In this method, early-modern Japanese printed book images are acquired and crawled to search for characters required as learning data. Then, the characters of the search result are checked by the user to see if they are correct or not to be registered in the database. We develop a web application that implements this method and evaluate it. As a result of the operation, a series of crawling processes are performed for each input character to register the character into the database as accurate training data. It is demonstrated that the collection of the required characters is more efficient than that of the conventional web application. In addition, the black pixel ratio of the collected character images is tabulated to calculate the distribution. Using this distribution, we implement a process to select a character image to be recognized by the black pixel ratio. The distribution of black pixel percentages is calculated using a current font, and it is confirmed that characters with high appearance frequency have a low black pixel percentage. Therefore, it is proved that the character selection process based on the black pixel ratio is effective in crawling for characters with low appearance frequency. In the future, the crawler application can be put into operation by introducing a sufficiently accurate character trimming and character recognition process. It is expected to efficiently search for and collect character types that have not been discovered before.

Acknowledgments This work is partially supported by Grant-in-Aid for Scientific Research from the Ministry of Education, Culture, Sports, Science, and Technology of Japan (MEXT) No. 20H04483.

References

1. C. Ishikawa, N. Ashida, Y. Enomoto, M. Takata, T. Kimesawa, K. Joe, Recognition of multi-fonts character in early-modern printed books, in PDPTA 2009, Vol. 2, pp. 728–734 (2009)
2. M. Fukuo, Y. Enomoto, N. Yoshii, M. Takata, T. Kimesawa, K. Joe, Evaluation of the SVM based multi-fonts kanji character recognition method for early-modern Japanese printed books, in PDPTA2011, Vol. 2, pp. 727–732 (2011)
3. T. Awazu, K. Kosaka, M. Takata, K. Joe, A multi-fonts kanji character recognition method for early-modern Japanese printed books. IPSJ Trans. on TOM **9**(2), 33–40 (2016)

4. National Diet Library Digital Collection: <http://dl.ndl.go.jp/>. Accessed 27 May 2020
5. S. Yasunami, N. Koiso, Y. Takemoto, Y. Ishikawa, M. Takata, K. Joe, Applying CNNs to early-modern printed Japanese character recognition, in *The 2019 International Conference on Parallel and Distributed Processing Techniques and Applications*, pp. 189–195 (2019)
6. G.K. Zipf, *Human Behavior & The Principle of Least Effort, An Introduction to Human Ecology* (Addison-Wesley Press Inc, 1949)
7. M. Fujita, Y. Takemoto, Y. Ishikawa, M. Takata, K. Joe, Collecting extremely low appearance characters from early-modern Japanese books, in *IPSJ SIGMPS, 2019-MPS-126(6)*, 1–6 (2019, written in Japanese)
8. K. Kosaka, T. Awazu, Y. Ishikawa, M. Takata, K. Joe, An effective and interactive training data collection method for early-modern Japanese printed character recognition, in *The 2015 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA2015)*, Vol. II, pp. 276–282 (2015)
9. Aozora Bunko. <https://github.com/aozorabunko/aozorabunko>. Accessed 27 May 2020
10. The Web framework for perfectionists with deadlines | Django. <https://www.djangoproject.com>. Accessed 27 May 2020
11. PostgreSQL. <https://www.postgresql.org>. Accessed 27 May 2020
12. Overview of the API Specification << About the National Diet Library Search. <https://iss.ndl.go.jp/information/api/riyou/>. Accessed 27 May 2020
13. National Diet Library Digital Search. <https://iss.ndl.go.jp>. Accessed 27 May 2020
14. Presentation API 2.1.1. <https://iiif.io/api/presentation/2.1/>. Accessed 27 May 2020
15. Image API 2.1.1. <https://iiif.io/api/image/2.1/>. Accessed 27 May 2020
16. IIIF | International Image Interoperability Framework. <https://iiif.io>. Accessed 27 May 2020
17. S. Iida, Y. Takemoto, Y. Ishikawa, M. Takata, K. Joe, Layout analysis using semantic segmentation for Imperial Meeting Minutes, in *The 2019 International Conference on Parallel and Distributed Processing Techniques and Applications*, pp. 135–141 (2019)

Application of the Orthogonal QD Algorithm with Shift to Singular Value Decomposition for Large Sparse Matrices



Hiroki Tanaka, Taiki Kimura, Tetsuaki Matsunawa, Shoji Mimotogi,
Masami Takata, Kinji Kimura, and Yoshimasa Nakamura

1 Introduction

Although a lithography simulation model is an essential technique for recent semiconductor manufacturing process, the model formula defined as a linear system is known as an ill-posed problem [9]. Regularization techniques are generally used to solve ill-posed problems. Nevertheless the question of how to choose the effective regularization matrix remains open when a general outline of solutions is known as prior information. In our paper [9], we introduced a method to add expected constraint to the solution by applying regularization technique with preconditioning matrix. By this regularization technique, an accurate simulation model can be achieved because the minimum norm least-squares solution using the truncated singular value decomposition from a few larger singular values becomes a reasonable solution based on physically appropriate prior knowledge. The augmented implicitly restarted Lanczos bidiagonalization (AIRLB) algorithm [3, 5] is suitable for the purpose of the truncated singular value decomposition from a few larger singular

H. Tanaka · Y. Nakamura
Kyoto University, Kyoto, Japan
e-mail: ynaka@i.kyoto-u.ac.jp

T. Kimura · T. Matsunawa · S. Mimotogi
KIOXIA Corporation, Yokohama, Japan
e-mail: taiki2.kimura@kioxia.com

M. Takata (✉)
Nara Women's University, Nara, Japan
e-mail: takata@ics.nara-wu.ac.jp

K. Kimura
Fukui University, Fukui, Japan
e-mail: kkimur@u-fukui.ac.jp

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_50

values of large-scale dense matrices. Thus, the AIRLB algorithm is important for obtaining the solution in lithography simulation modeling. We note that the AIRLB algorithm can also be applied to large sparse matrices.

The AIRLB algorithm is a truncated singular value decomposition algorithm. It is a Krylov subspace method and improves the Golub–Kahan–Lanczos (GKL) algorithm [7]. With the use of the restart method, the AIRLB algorithm requires less memory and computation time. In AIRLB, the singular value decomposition of a small matrix is performed in each iteration, and the QR algorithm [6] is conventionally used. However, the accuracy of the singular values obtained by the QR algorithm may not be satisfactory.

In this paper, we propose improvements to increase the accuracy of the AIRLB algorithm. Specifically, we implement the improvement of AIRLB by Ishida et al. [8]. Furthermore, instead of using QR, we use the orthogonal-qd-with-shift (OQDS) algorithm [1, 10] for the singular value decomposition of the inner small matrix, thereby providing high accuracy in terms of the singular-value errors.

2 Truncated Singular Value Decomposition

In the truncated singular value decomposition, an $m \times n$ ($m \geq n$) rectangular matrix $A \in \mathbb{R}^{m \times n}$, whose rank is r , is decomposed to $A \approx U \Sigma V^\top$. We set L ($L \leq r$) to the number of desired larger singular values. Here, the columns in the orthogonal matrices $U \in \mathbb{R}^{m \times L}$ and $V \in \mathbb{R}^{n \times L}$ consist of left and right singular vectors, respectively, and the diagonal matrix $\Sigma \in \mathbb{R}^{L \times L}$ has singular values as diagonal elements. These matrices satisfy the following conditions:

$$A \mathbf{v}_i = \sigma_i \mathbf{u}_i, \quad A^\top \mathbf{u}_i = \sigma_i \mathbf{v}_i \quad (i = 1, \dots, L), \quad (1)$$

$$U := [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_L] \in \mathbb{R}^{m \times L}, \quad V := [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_L] \in \mathbb{R}^{n \times L}, \quad (2)$$

$$\Sigma := \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_L) \in \mathbb{R}^{L \times L}, \quad \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_L > 0 \quad (3)$$

Here, the left and right singular vectors corresponding to the i th singular value σ_i are \mathbf{u}_i and \mathbf{v}_i , respectively.

3 Preconditioning Technique

In this paper, we consider the following system of m linear equations with n unknowns:

$$A \mathbf{x} = \mathbf{b}, \quad (4)$$

where $A \in \mathbb{R}^{m \times n}$ is a matrix, $\mathbf{x} \in \mathbb{R}^n$ denotes the solution vector, and $\mathbf{b} \in \mathbb{R}^m$ represents measured data. The goal is to determine an accurate approximation of x that minimizes an error (or discrepancy) function, $\|A\mathbf{x} - \mathbf{b}\|^2$. When the singular values of the matrix A gradually decay to zero, a common approach to determine a meaningful approximation of x is to employ Tikhonov regularization which can be written by a penalized least-squares problem of the form

$$\mathbf{x} = \operatorname{argmin}_x [\lambda \|C\mathbf{x}\|^2 + \|A\mathbf{x} - \mathbf{b}\|^2], \tag{5}$$

where C is the regularization matrix and λ is the regularization parameter. Although some regularization methods have been invented, such as the identity matrix or the Laplace operator, the question of how to choose the effective regularization matrix remains open when a general outline of solutions is known as prior information. Thus we introduced a regularization method to effectively utilize prior knowledge on both the observation condition and the physical background.

The basic concept of the introduced method is to regularize the solution by molding them into the form of an estimated physical outline of solutions. To realize this, the introduced preconditioning technique consists of two processes, scaling and uniformization. Suppose the solution of the penalized least-squares problem can be written by

$$\mathbf{x} = \alpha \mathbf{k} + \mathbf{\Delta}, \tag{6}$$

where α is the constant value corresponding to the scaling factor, $\mathbf{k} = (k_1, k_2, \dots, k_n)^T$ is the outline vector based on prior knowledge, and $\mathbf{\Delta}$ is the difference vector. We introduce a scaling matrix S as follows:

$$S = \operatorname{diag}(k_1, k_2, \dots, k_n). \tag{7}$$

We also give a Toeplitz matrix T whose elements are given by a Gaussian of the form

$$T = \begin{pmatrix} g_1 & g_n & \dots & g_3 & g_2 \\ g_2 & g_1 & \dots & g_4 & g_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ g_{n-1} & g_{n-2} & \dots & g_1 & g_n \\ g_n & g_{n-1} & \dots & g_2 & g_1 \end{pmatrix}, \quad g(x) = \frac{1}{\sqrt{2\pi s^2}} \exp\left(-\frac{x^2}{2s^2}\right), \tag{8}$$

where the mean and variance are 0 and s^2 , respectively. Here we define the right preconditioning matrix P using S and T as follows:

$$P = ST. \tag{9}$$

When P is added to the coefficient matrix A , Eq. (4) can be written as

$$A'x' = b, \tag{10}$$

where $A' = AP$ and $x' = P^{-1}x$. Seeking the minimum norm least-squares solution using the truncated singular value decomposition, x' is given by

$$x' = \sum_{i=1}^t \frac{(u_i^T b)}{\sigma_i} v_i, \tag{11}$$

where u, σ, v , and t are the left singular vector, singular value, right singular vector, and truncate number, respectively. Finally, the solution can be written in the form

$$x = Px' = \sum_{i=1}^t \frac{(u_i^T b)}{\sigma_i} P v_i. \tag{12}$$

Here we consider the singular value decomposition of $A' = AP = (AS)T$ in Eq. (10). If the variance of the Gaussian $s^2 \rightarrow \infty$, we shall assume the elements of A' becomes constants by uniformizing row components using T as follows:

$$A'_{s^2 \rightarrow \infty} = \begin{pmatrix} c_1 & c_1 & \dots & c_1 \\ c_2 & c_2 & \dots & c_2 \\ \vdots & \vdots & \ddots & \vdots \\ c_m & c_m & \dots & c_m \end{pmatrix} = (c_1, c_2, \dots, c_m)^T (1, 1, \dots, 1). \tag{13}$$

By using the singular value decomposition of $A'_{s^2 \rightarrow \infty}$, it follows that the right singular vector can be written by

$$v_1 = \frac{1}{\sqrt{n}} (1, 1, \dots, 1)^T. \tag{14}$$

Thus, the solution truncated with the maximum singular value is given by

$$x_{t=1} = \frac{(u_1^T b)}{\sigma_1} P v_1 = \frac{(u_1^T b)}{\sigma_1} S T v_1. \tag{15}$$

Focusing on $ST v_1$, although the elements of v_1 are almost uniformized, $T v_1$ goes to constant since it is further uniformized by T again. Similarly $(u_1^T b)/\sigma_1$ becomes constant, and therefore $x_{t=1}$ can be written in the form

$$x_{t=1} \approx \alpha k, \tag{16}$$

where α is constant. When s^2 is sufficiently large and αk is close to the true solution, an approximated solution with small relative residual norms can be obtained even though the value of t is small, and hence

$$\mathbf{x} = \alpha \mathbf{k} + \sum_{i=2}^{t \ll n} \frac{(\mathbf{u}_i^T \mathbf{b})}{\sigma_i} P \mathbf{v}_i. \quad (17)$$

From this equation, we can see that it is able to determine reasonable solution by using the AIRLB algorithm. In contrast, if $\alpha \mathbf{k}$ is far from the true solution, the above solution becomes the following:

$$\mathbf{x} = \alpha \mathbf{k} + \sum_{i=2}^{t \sim n} \frac{(\mathbf{u}_i^T \mathbf{b})}{\sigma_i} P \mathbf{v}_i, \quad (18)$$

This means that the value of t must be large enough to obtain small relative residual norms.

4 Algorithms for Truncated Singular Value Decomposition

4.1 *Augmented Implicitly Restarted Lanczos Bidiagonalization Algorithm*

The AIRLB algorithm is a Krylov subspace method that improves the GKL algorithm by using the restart method. In the GKL method, the Krylov subspace is expanded at each iteration, and new basis vectors are added until an approximation matrix is obtained. Consequently, a large amount of memory and computation time is required for reorthogonalization. Thus, to reduce the cost of reorthogonalization, the basis number in the Krylov subspace should be limited. Using the information of the restricted subspace, we generate the initial vector that is used in the next information. This operation is generally termed as restart. The AIRLB algorithm applies the restart method to GKL and can obtain the singular triplets corresponding to large singular values of large matrices faster than GKL and with the use of less memory. Let ℓ be the number of desired singular triplets, and let k be the basis number of the Krylov subspace. In general, the basis number of the Krylov subspace k is about twice the number of desired singular triplets ℓ . This algorithm is given in Algorithm 1.

5 Improvements

Herein, certain improvements of the AIRLB algorithm are described. We first describe the improvement by Ishida et al. [8]. Subsequently, we introduce the OQDS algorithm for the restart operation.

5.1 AIRLB Algorithm Using the Improvement by Ishida et al.

In AIRLB, the singular value decomposition of the small matrix \tilde{B}_k is performed internally, and the result is used at the restarting point of the algorithm. If computation errors are not considered, the obtained singular vectors are orthogonal. However, with computation errors, the singular vectors may not be orthogonal. To avoid this, by using the QR decomposition with Householder reflector, Ishida et al. [8] developed an improved algorithm that restarts with orthogonalization of both sides of the singular vectors of \tilde{B}_k . In this study, instead of the Householder reflector, we use the modified Gram–Schmidt algorithm to increase speed.

Algorithm 1 AIRLB algorithm

```

1: Set an  $n$ -dimensional unit vector  $\tilde{\mathbf{v}}_1, i \leftarrow 1$ 
2: repeat
3:    $\tilde{P}_i \leftarrow [\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \dots, \tilde{\mathbf{v}}_i]$ 
4:   while  $i \leq k$  do
5:      $\mathbf{u} \leftarrow A\tilde{\mathbf{v}}_i, \text{Reorthogonalization}(\tilde{Q}_i, \mathbf{u}), \tilde{\alpha}_i \leftarrow \|\mathbf{u}\|, \tilde{\mathbf{u}}_i \leftarrow \mathbf{u}/\tilde{\alpha}_i$ 
6:      $\tilde{Q}_i \leftarrow [\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2, \dots, \tilde{\mathbf{u}}_i], \mathbf{v} \leftarrow A^\top \tilde{\mathbf{u}}_i, \text{Reorthogonalization}(\tilde{P}_i, \mathbf{v})$ 
7:      $\tilde{\beta}_i \leftarrow \|\mathbf{v}\|, \tilde{\mathbf{v}}_{i+1} \leftarrow \mathbf{v}/\tilde{\beta}_i, \tilde{P}_{i+1} \leftarrow [\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \dots, \tilde{\mathbf{v}}_{i+1}], i \leftarrow i + 1$ 
8:   end while
9:    $\tilde{\mathbf{v}}_{\ell+1} \leftarrow \tilde{\mathbf{v}}_{k+1}$ 
10:  Compute the singular value decomposition of  $\tilde{B}_k = \tilde{U}_k \tilde{\Sigma}_k \tilde{V}_k^\top$ 
11:  for  $i = 1, \dots, \ell$  do
12:     $\tilde{\rho}_i \leftarrow \tilde{\beta}_k \tilde{\mathbf{u}}_i(k)$ 
13:  end for
14:   $\tilde{B}_k(1 : \ell, 1 : \ell) \leftarrow \tilde{\Sigma}_k(1 : \ell, 1 : \ell), \tilde{Q}_k \leftarrow \tilde{Q}_k \tilde{U}_k(:, 1 : \ell), \tilde{P}_k \leftarrow \tilde{P}_k \tilde{V}_k(:, 1 : \ell)$ 
15:   $i \leftarrow \ell + 1$ 
16: until  $\max_{1 \leq i \leq \ell} \frac{|\tilde{\rho}_i|}{\sqrt{2}} \leq \delta$  (threshold value)
17:  $\tilde{\mathbf{u}}_i \leftarrow \tilde{Q}_k(:, i), \tilde{\mathbf{v}}_i \leftarrow \tilde{P}_k(:, i)$ 
18: Output  $(\tilde{\sigma}_i, \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i)$  for  $i = 1, \dots, \ell$ 

```

The improved AIRLB algorithm adopts the QR decomposition using the modified Gram–Schmidt process with $\tilde{V}_\ell = Q_v R_v$ to orthogonalize \tilde{V}_ℓ . Let the orthogonal matrix Q_v be a new \tilde{V}_ℓ :

$$\tilde{V}_\ell \leftarrow [\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \dots, \tilde{\mathbf{v}}_\ell], \quad \tilde{V}_\ell = Q_v R_v, \quad \tilde{V}_\ell \leftarrow Q_v. \quad (19)$$

The left singular vectors \tilde{U}_ℓ can be orthogonalized in the same manner as \tilde{V}_ℓ :

$$\tilde{U}_\ell \leftarrow [\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2, \dots, \tilde{\mathbf{u}}_\ell], \quad \tilde{U}_\ell = Q_u R_u, \quad \tilde{U}_\ell \leftarrow Q_u. \quad (20)$$

This improvement increases the orthogonality of \tilde{V}_ℓ and \tilde{U}_ℓ ; however, \tilde{V}_ℓ and \tilde{U}_ℓ may not satisfy the following equations:

$$\tilde{B}_k \tilde{V}_\ell = \tilde{U}_\ell \tilde{\Sigma}_\ell, \quad \tilde{B}_k^\top \tilde{U}_\ell = \tilde{V}_\ell \tilde{\Sigma}_\ell. \quad (21)$$

To satisfy Eq. (21) approximately, Ishida et al. use the Rayleigh quotient [11], which is defined as $\tilde{\Sigma}_k(i, i) = \tilde{u}_i^\top \tilde{B}_k \tilde{v}_i$. Thereby, Eqs. (21) are satisfied approximately. This algorithm is given in Algorithm 2.

5.2 Orthogonal-qd-with-Shift Algorithm

5.2.1 Outline

The OQDS method [10] is suitable for the computation of singular values with high accuracy in terms of relative errors. Let $L^{(i)}$ be an $n \times n$ lower bidiagonal matrix and $U^{(i)}$ be an $n \times n$ upper bidiagonal matrix:

Algorithm 2 AIRLB algorithm using the improvement by Ishida et al.

```

1: Set an  $n$ -dimensional unit vector  $\tilde{v}_1, i \leftarrow 1$ 
2: repeat
3:    $\tilde{P}_i \leftarrow [\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_i]$ 
4:   while  $i \leq k$  do
5:      $\mathbf{u} \leftarrow A \tilde{v}_i$ , Reorthogonalization( $\tilde{Q}_i, \mathbf{u}$ ),  $\tilde{\alpha}_i \leftarrow \|\mathbf{u}\|$ ,  $\tilde{\mathbf{u}}_i \leftarrow \mathbf{u} / \tilde{\alpha}_i$ 
6:      $\tilde{Q}_i \leftarrow [\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2, \dots, \tilde{\mathbf{u}}_i]$ ,  $\mathbf{v} \leftarrow A^\top \tilde{\mathbf{u}}_i$ , Reorthogonalization( $\tilde{P}_i, \mathbf{v}$ )
7:      $\tilde{\beta}_i \leftarrow \|\mathbf{v}\|$ ,  $\tilde{v}_{i+1} \leftarrow \mathbf{v} / \tilde{\beta}_i$ ,  $\tilde{P}_{i+1} \leftarrow [\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_{i+1}]$ ,  $i \leftarrow i + 1$ 
8:   end while
9:    $\tilde{v}_{\ell+1} \leftarrow \tilde{v}_{k+1}$ 
10:  Compute the singular value decomposition of  $\tilde{B}_k = \tilde{U}_k \tilde{\Sigma}_k \tilde{V}_k^\top$ 
11:  Compute the QR decomposition  $\tilde{V}_\ell = Q_v R_v$  using the modified Gram-Schmidt algorithm
12:   $\tilde{V}_\ell \leftarrow Q_v$ 
13:  Compute the QR decomposition  $\tilde{U}_\ell = Q_u R_u$  using the modified Gram-Schmidt algorithm
14:   $\tilde{U}_\ell \leftarrow Q_u$ 
15:   $\tilde{\Sigma}_k(i, i) \leftarrow \tilde{u}_i^\top \tilde{B}_k \tilde{v}_i$  for  $i = 1, \dots, \ell$ 
16:  for  $i = 1, \dots, \ell$  do
17:     $\tilde{\rho}_i \leftarrow \tilde{\beta}_k \tilde{u}_i(k)$ 
18:  end for
19:   $\tilde{B}_k(1 : \ell, 1 : \ell) \leftarrow \tilde{\Sigma}_k(1 : \ell, 1 : \ell)$ ,  $\tilde{P}_k \leftarrow \tilde{P}_k \tilde{V}_\ell$ ,  $\tilde{Q}_k \leftarrow \tilde{Q}_k \tilde{U}_\ell$ ,  $i \leftarrow \ell + 1$ 
20: until  $\max_{1 \leq i \leq \ell} \frac{|\tilde{\rho}_i|}{\sqrt{2}} \leq \delta$  (threshold value)
21:  $\tilde{\mathbf{u}}_i \leftarrow \tilde{Q}_k(:, i)$ ,  $\tilde{\mathbf{v}}_i \leftarrow \tilde{P}_k(:, i)$ 
22: Output  $(\tilde{\sigma}_i, \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i)$  for  $i = 1, \dots, \ell$ 

```

$$L^{(i)} = \begin{pmatrix} \alpha_1^{(i)} & & & & \\ \beta_1^{(i)} & \alpha_2^{(i)} & & & \\ & \ddots & \ddots & & \\ & & \beta_{n-1}^{(i)} & \alpha_n^{(i)} & \\ & & & & \end{pmatrix}, \quad U^{(i)} = \begin{pmatrix} \gamma_1^{(i)} & \zeta_1^{(i)} & & & \\ & \gamma_2^{(i)} & \ddots & & \\ & & \ddots & \ddots & \\ & & & \ddots & \zeta_{n-1}^{(i)} \\ & & & & \gamma_n^{(i)} \end{pmatrix}. \quad (22)$$

In the OQDS algorithm, the following three operations are repeated for $L^{(i)}$ and $U^{(i)}$:

1. Compute a shift $u^{(i)}$ satisfying $0 \leq u^{(i)} \leq \sigma_{\min}(L^{(i)})$.
2. LU step

$$P^{(i)} \begin{pmatrix} L^{(i)} \\ t^{(i)} I_n \end{pmatrix} = \begin{pmatrix} U^{(i)} \\ t^{(i+1)} I_n \end{pmatrix}, \quad t^{(i+1)} = \sqrt{(t^{(i)})^2 + (u^{(i)})^2} \quad (23)$$

3. UL step

$$\begin{pmatrix} I_n & O \\ O & (Q^{(i)})^\top \end{pmatrix} \begin{pmatrix} U^{(i)} \\ t^{(i+1)} I_n \end{pmatrix} Q^{(i)} = \begin{pmatrix} L^{(i+1)} \\ t^{(i+1)} I_n \end{pmatrix}. \quad (24)$$

$P^{(i)}$ and $Q^{(i)}$ are $2n \times 2n$ and $n \times n$ orthogonal matrices, respectively. $P^{(i)}$ is computed by using the Givens and the generalized Givens rotation [10]. $Q^{(i)}$ consists of Givens rotations. $t^{(i_0)} I_n$ is a diagonal matrix with the same value. Let Σ be a diagonal matrix arranged in descending order of singular values. When $L^{(i_0)}$ converges to a diagonal matrix, if the split operation does not occur, then $\Sigma(k, k) = \sqrt{(L^{(i_0)}(k, k))^2 + (t^{(i_0)})^2}$ ($k = 1, \dots, n$). To obtain right singular vectors, an orthogonal matrix V is computed with $V = Q^{(0)} \dots Q^{(i_0-1)}$. Using a Givens rotation, we add $t^{(i_0)} I_n$ to $L^{(i_0)}$. Then, $L^{(i_0)}$ and $t^{(i_0)} I_n$ become Σ and the zero matrix, respectively. U' is defined as follows:

$$U' = (I_n \ O) (P^{(0)})^\top \begin{pmatrix} I_n & O \\ O & Q^{(0)} \end{pmatrix} \dots (P^{(i_0-1)})^\top \begin{pmatrix} I_n & O \\ O & Q^{(i_0-1)} \end{pmatrix} (P^{(i_0)})^\top, \quad (25)$$

where $(P^{(i_0)})^\top$ is used to add $t^{(i_0)} I_n$ to $L^{(i_0)}$. U' is split into two parts as follows:

$$U' = (U_{11} \ U_{12}). \quad (26)$$

The $n \times n$ orthogonal matrix U_{11} consists of left singular vectors.

5.2.2 LU and UL Steps

The LU step is an operation that transforms $L^{(i)}$ into $U^{(i)}$.

Algorithm 4 UL step

```

1: Set  $\eta_1^{(i)} := \gamma_1^{(i)}$ ;
2: for  $k := 1, 2, \dots, n-1$  do
3:   Set  $\alpha_k^{(i+1)} := \sqrt{(\eta_k^{(i)})^2 + (\zeta_k^{(i)})^2}$ ;
4:   if  $\alpha_k^{(i+1)} = 0$  then
5:     Set  $\beta_k^{(i+1)} := 0$ ,  $\eta_{k+1}^{(i)} := \gamma_{k+1}^{(i)}$ ;
6:   else
7:     Set  $\beta_k^{(i+1)} := (\zeta_k^{(i)} / \alpha_k^{(i+1)}) \gamma_{k+1}^{(i)}$ ;
8:     Set  $\eta_{k+1}^{(i)} := (\eta_k^{(i)} / \alpha_k^{(i+1)}) \gamma_{k+1}^{(i)}$ ;
9:   end if
10: end for
11: Set  $\alpha_n^{(i)} := \eta_n^{(i)}$ ;

```

The operation in Eq.(28) can be performed using a Givens rotation from the right-hand side. The UL step, with the elements of $U^{(i)}$ and $L^{(i+1)}$, is shown in Algorithm 4.

In the OQDS algorithm, the Givens and the generalized Givens rotation are used in the LU and UL steps. Usually, in the implementation, the Givens rotation can be computed by using `xROTG`, which is a level-1 routine in BLAS [4]. However, the high-precision computation of the Givens rotation that is required in the OQDS algorithm should be performed using `xLARTG` in LAPACK.

6 Numerical Experiments

Herein, numerical experiments are conducted to evaluate the AIRLB algorithm.

6.1 Experimental Environment

The experiments are conducted in the following environment:

- CPU: Intel Xeon E5-2695 v4 (2.1 GHz \times 18 cores)
- Memory: 128 GB
- Compiler: ifort version 19.1.0.166
- Compiler options: `-O3 -ip -xHOST -fopenmp -fp-model precise`
- Library: Intel Math Kernel Library 2019.1.0

Here, a board has two CPUs.

We compare the following AIRLIB algorithms in terms of the number of iterations, the singular value errors, and the orthogonality errors of the singular vectors:

- AIRLB(O): The original AIRLB algorithm with QR.
- AIRLB(IO): The AIRLB algorithm with QR, using the improvement by Ishida et al.
- AIRLB(M): The AIRLB algorithm with QR in which the Givens rotation with high accuracy in [2] is adopted.
- AIRLB(IM): The AIRLB algorithm with QR in which the Givens rotation with high accuracy in [2] is adopted, using the improvement by Ishida et al.
- AIRLB(S): The AIRLB algorithm with OQDS.
- AIRLB(IS): The AIRLB algorithm with OQDS, using the improvement by Ishida et al.

AIRLB(O) is the algorithm in [3, 5]. For the AIRLB algorithm, it is necessary to set the number of the required singular triplets and the basis number of the Krylov subspace. In this experiment, the number of the required singular triplets is $\ell = 10$, and the basis number of the Krylov subspace is $k = 20$.

The results are obtained from the following two types of matrices as input. We first use real upper bidiagonal matrices $A_1 \in \mathbb{R}^{10,000 \times 10,000}$, $A_2 \in \mathbb{R}^{20,000 \times 20,000}$, $A_3 \in \mathbb{R}^{30,000 \times 30,000}$, $A_4 \in \mathbb{R}^{40,000 \times 40,000}$, and $A_5 \in \mathbb{R}^{50,000 \times 50,000}$. All diagonal and off-diagonal elements of A_j ($j = 1, \dots, 5$) are 1. The i -th singular values of these matrices are exactly known as follows:

$$\sigma_i = 2 \cos \left(\frac{i}{2n+1} \pi \right), \quad i = 1, 2, \dots, n, \quad (29)$$

where n is the matrix size. From Eq. (29), large singular values of A_j ($j = 1, \dots, 5$) are quite clustered around 2. Thus, the singular value decomposition of these input matrices is difficult to perform by the Krylov method. Subsequently, we use real random sparse matrices $A_6 \in \mathbb{R}^{1,000,000 \times 1,000,000}$, $A_7 \in \mathbb{R}^{1,500,000 \times 1,500,000}$, and $A_8 \in \mathbb{R}^{2,000,000 \times 2,000,000}$ as input. Each row has 1,000 elements, which are uniform random numbers in $[0, 1)$. The column indices of the 1,000 elements are also selected by random numbers. In addition, we use real random dense matrices $B_1 \in \mathbb{R}^{10,000 \times 10,000}$, $B_2 \in \mathbb{R}^{20,000 \times 20,000}$, $B_3 \in \mathbb{R}^{30,000 \times 30,000}$, $B_4 \in \mathbb{R}^{40,000 \times 40,000}$, and $B_5 \in \mathbb{R}^{50,000 \times 50,000}$ as input. All elements in each row consist of random values of $[0, 1)$. B_j ($j = 1, \dots, 5$) are examples of large dense matrices similar to the data matrices in lithography simulation modeling.

In these experiments, we use four metrics. Specifically,

$$\frac{1}{\ell} \sum_{1 \leq i \leq \ell} \frac{1}{\sqrt{2}} \sqrt{\|A \tilde{\mathbf{v}}_i - \tilde{\sigma}_i \tilde{\mathbf{u}}_i\|_2^2 + \|A^\top \tilde{\mathbf{u}}_i - \tilde{\sigma}_i \tilde{\mathbf{v}}_i\|_2^2}, \quad (30)$$

is the average error value, and

$$\max_{1 \leq i \leq \ell} \frac{1}{\sqrt{2}} \sqrt{\|A \tilde{\mathbf{v}}_i - \tilde{\sigma}_i \tilde{\mathbf{u}}_i\|_2^2 + \|A^\top \tilde{\mathbf{u}}_i - \tilde{\sigma}_i \tilde{\mathbf{v}}_i\|_2^2}, \quad (31)$$

is the maximum error value. Furthermore, the orthogonality errors of the singular vectors are evaluated using the Frobenius norm as follows:

$$\|\tilde{V}^T \tilde{V} - I\|_F, \quad \|\tilde{U}^T \tilde{U} - I\|_F, \quad (32)$$

where

$$\tilde{V} := [\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_\ell], \quad \tilde{U} := [\tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_\ell]. \quad (33)$$

The orthogonality error of the right and left singular vectors is computed using Eq. (32).

6.2 Experimental Results and Discussion

In the numerical experiments, we compare the AIRLB algorithms.

The computation results when the six AIRLB algorithms are applied to the matrices A_j ($j = 1, \dots, 8$) and B_j ($j = 1, \dots, 5$) are shown in Table 1.

It can be seen that the computation time and the number of iterations do not vary significantly among the algorithms. As our improvement is a low-level routine, its effect on the computation time and the number of iterations is quite small.

Moreover, AIRLB(IO), AIRLB(IM), and AIRLB(IS) outperform AIRLB(O), AIRLB(M), and AIRLB(S). Thus, the improvement by Ishida et al. is effective.

The results when the algorithms are applied to A_j ($j = 6, 7, 8$) imply that the orthogonality of AIRLB(IS) is almost the same as that of AIRLB(IO) and AIRLB(IM), whereas A_j ($j = 1, \dots, 5$) and B_j ($j = 1, \dots, 5$) imply that the orthogonality of AIRLB(IS) is slightly larger than that of AIRLB(IO) and AIRLB(IM). However, the error of AIRLB(IS) in Eqs. (30) and (31) is the smallest in A_j ($j = 1, \dots, 5$) and B_j ($j = 2, \dots, 5$). Particularly in A_j ($i = 6, 7, 8$), the error of AIRLB(IS) is extremely small. As the QR algorithm focuses on the orthogonalization of singular vectors, a vector is not always in the correct direction. By contrast, in the OQDS algorithm, as the singular values are highly accurate, it is highly likely that the singular vector has the correct direction. Thus, in the OQDS algorithm, the Rayleigh quotient is effective. Furthermore, the modified Gram–Schmidt algorithm reorthogonalizes the singular vectors to obtain vectors with high orthogonality. Hence, OQDS is suitable for AIRLB.

Therefore, to improve the accuracy of the AIRLB algorithm, the proposed method should be adopted (Table 2).

Table 1 Experimental results 1

	AIRLB(O)	AIRLB(IO)	AIRLB(M)	AIRLB(IM)	AIRLB(S)	AIRLB(IS)
Computation time [sec.]						
A ₁	1530.20	1503.16	1536.20	1634.59	1608.09	1572.80
A ₂	8069.37	8158.79	8034.20	8121.81	8416.41	8614.89
A ₃	19968.92	20379.46	19785.69	19576.08	20875.31	20706.92
A ₄	46524.93	46812.34	46540.92	46809.14	47975.10	49498.72
A ₅	74450.74	73874.38	75437.44	73279.87	78497.13	77062.57
A ₆	17981.46	17501.75	17791.94	17640.17	17922.25	17787.12
A ₇	106883.67	107158.10	107312.93	107749.28	108782.18	108924.68
A ₈	94702.95	94194.52	94051.48	94373.52	95913.45	96410.16
Number of iterations						
A ₁	220592	220592	220592	220592	222870	222870
A ₂	917671	917551	917671	917557	927097	926935
A ₃	1879030	1877226	1879028	1877268	1900160	1898486
A ₄	3753738	3759758	3753738	3759410	3807026	3798990
A ₅	5214598	5154612	5214250	5156050	5248380	5214596
A ₆	1426	1426	1426	1426	1447	1447
A ₇	5558	5558	5558	5558	5617	5617
A ₈	3598	3598	3598	3598	3654	3654
Average error in Eq. (30) [10^{-13}]						
A ₁	784.31	535.13	695.29	509.71	386.61	112.39
A ₂	3869.85	2038.85	3245.73	1990.75	1037.58	351.04
A ₃	11434.18	7143.60	10200.81	7176.34	2194.41	1121.46
A ₄	13370.29	7022.24	10353.82	6889.10	2190.06	1099.48
A ₅	28040.73	11939.45	16689.82	11521.03	3563.38	2087.90
A ₆	68.81	45.99	57.08	44.71	28.26	4.38
A ₇	75.59	30.60	66.10	31.07	68.10	6.07
A ₈	128.93	78.45	108.71	79.32	76.76	7.12
Maximum error in Eq. (31) [10^{-13}]						
A ₁	1908.57	1624.59	1943.78	1555.10	897.58	303.27
A ₂	9222.61	5803.51	9080.75	5580.26	2958.53	632.04
A ₃	30175.15	23989.11	26090.35	23688.93	6072.40	3510.45
A ₄	27836.38	18815.29	20941.14	18692.44	4917.79	1925.61
A ₅	48507.90	25201.63	34641.43	24316.35	7054.70	6104.63
A ₆	104.64	80.56	89.92	80.28	69.46	6.00
A ₇	139.70	63.40	117.10	63.48	148.74	11.12
A ₈	206.92	179.38	171.75	177.39	169.19	11.73

(continued)

Table 1 (continued)

Orthogonality of V [10^{-13}]						
A_1	1288.14	55.97	774.78	59.07	672.44	43.84
A_2	4080.77	293.88	1373.61	297.08	1175.92	263.51
A_3	12021.72	410.75	2887.08	420.76	2354.18	597.61
A_4	15405.99	496.76	3405.67	534.67	2330.48	755.49
A_5	28959.88	1148.29	2997.29	1169.57	3437.23	1477.24
A_6	8.32	0.06	5.85	0.16	5.44	0.15
A_7	14.80	0.31	11.02	0.38	12.17	0.15
A_8	19.47	0.17	13.28	0.29	12.39	0.26
Orthogonality of U [10^{-13}]						
A_1	1576.93	51.64	759.38	58.12	792.38	44.18
A_2	7748.00	268.12	1351.04	299.05	1483.35	268.82
A_3	27995.79	429.63	2868.09	426.47	3000.45	615.29
A_4	36633.87	576.41	3389.27	537.07	2635.61	790.69
A_5	86463.98	1307.91	3012.84	1173.93	4708.67	1520.71
A_6	3.04	0.11	6.26	0.14	6.15	0.15
A_7	10.47	0.24	10.80	0.34	12.05	0.37
A_8	8.93	0.19	13.20	0.22	14.14	0.29

Table 2 Experimental results 2

	AIRLB(O)	AIRLB(IO)	AIRLB(M)	AIRLB(IM)	AIRLB(S)	AIRLB(IS)
Computation time [sec.]						
B_1	15.06	15.11	15.10	15.38	15.01	15.56
B_2	104.97	105.80	99.78	106.18	106.38	103.08
B_3	466.93	467.64	481.73	478.75	466.34	476.42
B_4	638.28	642.97	657.44	658.77	646.82	659.99
B_5	1882.91	1861.32	1897.79	1893.48	1906.20	1919.22
Number of iterations						
B_1	88	88	88	88	89	89
B_2	195	195	195	195	197	197
B_3	414	414	414	414	418	418
B_4	323	323	323	323	328	328
B_5	608	608	608	608	617	617
Average error in Eq. (30) [10^{-13}]						
B_1	10.89	9.14	10.89	8.61	9.14	3.66
B_2	23.49	17.38	21.73	17.60	17.23	6.10
B_3	123.82	114.94	123.71	115.06	24.70	9.52
B_4	98.94	75.23	97.19	76.02	44.96	14.48
B_5	80.47	67.11	80.95	65.91	37.70	18.04

(continued)

Table 2 (continued)

Maximum error in Eq. (31) [10^{-13}]						
B_1	18.30	13.50	16.97	12.44	33.19	13.27
B_2	35.77	36.27	36.71	36.27	54.69	25.56
B_3	935.74	935.87	935.87	935.87	57.53	41.10
B_4	548.76	545.57	544.80	542.95	89.77	64.16
B_5	350.32	350.19	350.37	350.25	80.31	76.07
Orthogonality of V [10^{-14}]						
B_1	6.71	0.17	5.49	0.24	5.68	0.13
B_2	10.33	0.26	9.45	0.34	9.13	0.20
B_3	16.98	0.28	12.35	0.38	12.88	0.50
B_4	24.37	0.41	16.59	0.45	15.31	0.40
B_5	16.31	0.61	7.46	0.46	9.74	0.60
Orthogonality of U [10^{-14}]						
B_1	4.69	0.14	3.92	0.18	4.85	0.25
B_2	8.58	0.20	7.42	0.22	6.79	0.20
B_3	17.26	0.27	13.74	0.38	12.18	0.38
B_4	21.09	0.21	13.88	0.29	17.78	0.29
B_5	13.70	0.58	11.25	0.77	10.41	0.58

7 Conclusion

In this paper, we adopted the OQDS algorithm in AIRLB using the improvement by Ishida et al.

The numerical experiments demonstrated that, in terms of the singular value errors and the orthogonality errors of the singular vectors, the AIRLB algorithm using the improvement by Ishida et al. was more accurate with OQDS than with QR, when random dense matrices similar to data matrices in lithography simulation modeling were used as input. That is, the proposed method improves the truncated singular value decomposition in a large matrix, resulting in obtaining more accurate solutions of lithography simulation modeling. In terms of the computation time and number of iterations, no significant differences were observed in any of the test cases.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number JP17H02858 and JP17K00167.

References

1. S. Araki, H. Tanaka, M. Takata, K. Kimura, Y. Nakamura, Fast computation method of column space by using the DQDS method and the OQDS method, in *Proceedings of PDPTA 2018* (2018), pp. 333–339
2. S. Araki, M. Takata, K. Kimura, Y. Nakamura, On an implementation of two-sided Jacobi method, in *Proceedings of PDPTA 2019* (2019), pp. 156–162

3. J. Baglama, L. Reichel, Augmented implicitly restarted Lanczos bidiagonalization methods. *SIAM J. Sci. Comput.* **27**(1), 19–42 (2005)
4. Basic Linear Algebra Subprograms, Netlib. <http://netlib.org/blas/>. Last Accessed 31 Mar 2020
5. D. Calvetti, L. Reichel, D.C. Sorensen, An implicitly restarted Lanczos method for large symmetric eigenvalue problems. *Electron. Trans. Numer. Anal.* **2**(1), 1–21 (1994)
6. J. Demmel, W. Kahan, Accurate singular values of bidiagonal matrices. *SIAM J. Sci. Stat. Comput.* **11**(5), 873–912 (1990)
7. G.H. Golub, W. Kahan, Calculating the singular values and pseudo-inverse of a matrix. *J. SIAM Ser. B Numer. Anal.* **2**(2), 205–224 (1965)
8. Y. Ishida, M. Takata, K. Kimura, Y. Nakamura, Improvement of the augmented implicitly restarted Lanczos bidiagonalization method in single precision floating point arithmetic. *IPJSJ Trans. Model. Appl.* **11**, 19–25 (2018)
9. T. Kimura, T. Matsunawa, S. Mimotogi, Regularized minimum norm least squares solution using preconditioning technique for semiconductor process modeling. *IPJSJ Trans. Model. Appl.* **12**, 26–36 (2019)
10. U. von Matt, The orthogonal qd-algorithm. *SIAM J. Sci. Comput.* **18**, 1163–1186 (1997)
11. B.N. Parlett, The symmetric Eigenvalue problem. SIAM, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, ISBN 978-0-898714-02-9; **10**, 24–38 (1998)

On an Implementation of the One-Sided Jacobi Method with High Accuracy



Masami Takata, Sho Araki, Kinji Kimura, and Yoshimasa Nakamura

1 Introduction

Many mathematical applications require a generalized eigenvalue formula that comprises a symmetric matrix and positive definite symmetric matrix. However, they use only some eigenvalues and the corresponding eigenvectors. The Sakurai–Sugiura method [15], also called truncated eigenvalue decomposition, uses a column space, which is computed by decomposing a rectangular matrix via singular value decomposition. Generally, a given matrix is transformed into a bidiagonal one using a Householder transformation [3] as a preprocessing method. In [1], a computation method for column space, adopted to a bidiagonal matrix, was proposed. The method combined the differential qd algorithm with shifts [7, 12] and orthogonal qd algorithm with shifts (OQDS) [11]. The Sakura–Sugiura method requires only a column space, which is based on left singular vectors, in a given upper bidiagonal matrix. Because the row space in a lower bidiagonal matrix is equal to the column space in a upper bidiagonal matrix, the row space can be computed using right singular vectors, achieved by employing the OQDS method, which was proposed in [1].

M. Takata (✉)
Nara Women's University, Nara, Japan
e-mail: takata@ics.nara-wu.ac.jp

S. Araki · Y. Nakamura
Kyoto University, Kyoto, Japan
e-mail: ynaka@i.kyoto-u.ac.jp

K. Kimura
Fukui University, Fukui, Japan
e-mail: kkimur@u-fukui.ac.jp

In [1], the test matrices were bidiagonal. However, in the Sakurai–Sugiura method, the singular value decomposition of an upper triangular matrix, whose dimension is small, is required. In the case of bidiagonalization using the Householder transformation, a rounding error may occur computationally. Thus, an algorithm which does not employ the Householder transformation should be used.

The Jacobi method for performing singular value decomposition can compute all singular values and singular vectors. James Demmel and Kresimir Veselic reported that the Jacobi method was more accurate than the QR method [4]; additionally, the computation cost for the former was insignificant for comparatively small matrices. One- and two-sided Jacobi methods have been proposed as the implementations of the Jacobi method for singular value decomposition [2, 5, 6, 8, 9]. They aim for computation accuracy and speed, respectively. In this study, to perform singular value decomposition with high accuracy, we propose a novel implementation of the one-sided Jacobi method, whose conventional implementation has theoretically high accuracy. Practically, there exist cases wherein singular vectors with sufficient orthogonality cannot be computed. To avoid this problem, in the proposed implementation, a Givens rotation with high accuracy and the fused multiply-accumulate are adopted. We confirmed that our implemented one-sided Jacobi method has higher accuracy than that of the one-sided Jacobi method implemented in Linear Algebra PACKage (LAPACK) [10].

Section 2 introduces a conventional implementation of the one-sided Jacobi method. In Sect. 3, we prepare the Givens rotation with high accuracy. Section 4 proposes a novel implementation of the one-sided Jacobi method. Finally, in Sect. 5, we compare the proposed implementation with the subroutine in LAPACK.

2 Conventional Implementation for the One-Sided Jacobi Method

Algorithm 1 shows a pseudocode of the one-sided Jacobi method [4]. It requires an $m \times n$ ($m \geq n$) real matrix A and a convergence-judgment threshold tol and ensures Σ , U , and V . The terms Σ , U , and V denote the matrices whose elements are singular values, left singular vectors, and right singular vectors in A , respectively. Generally, tol is set to $tol = \sqrt{m}\varepsilon$, where ε denotes a machine epsilon, which is adopted in `xGESVJ` implemented in LAPACK, considering the error in the inner-product computation.

The algorithm comprises a double loop with the main part. The outer loop repeats until the result of the inner loop converges. The inner loop performs through a subscript *pairs*, which is given using subroutine `jacobi_pairs`. This generates a pair that contains at least one pair of all the integers from 1 to n . Thus, *pairs* contain at least $n(n - 1)/2$ entries.

3 Givens Rotation with High Accuracy

Jacobi rotation in the Jacobi method involves the same computation as in Givens rotation. Givens rotation is performed using vectors \mathbf{x} and \mathbf{y} as follows:

$$\mathbf{x} \leftarrow \cos(\theta)\mathbf{x} + \sin(\theta)\mathbf{y}, \quad \mathbf{y} \leftarrow -\sin(\theta)\mathbf{x} + \cos(\theta)\mathbf{y}. \tag{1}$$

Algorithm 1 Pseudocode of the one-sided Jacobi method

```

Require:  $A = [a_1 \ a_2 \ \dots \ a_n], tol$ 
Ensure:  $(U, \Sigma, V)$ 
1:  $V := [v_1 \ v_2 \ \dots \ v_n] := I_{n,n}$ 
2: repeat
3:    $maxt := 0$ 
4:    $pairs := \text{jacobipairs}()$ 
5:   for  $(i, j)$  in  $pairs$  do
6:      $x := a_i^\top a_i$ 
7:      $y := a_j^\top a_j$ 
8:      $g := a_i^\top a_j$ 
9:      $\tau := |g|/\sqrt{x \times y}$ 
10:     $maxt := \max(maxt, \tau)$ 
11:    if  $\tau > tol$  then
12:      Computation of Jacobi rotation
13:       $f := (x - y)/2$ 
14:       $t := g / (f + \text{sign}(\sqrt{g^2 + f^2}, f))$ 
15:       $r := \sqrt{1 + t^2}$ 
16:       $c := 1/r$ 
17:       $s := t/r$ 
18:      Effect of Jacobi rotation
19:       $q := a_i$ 
20:       $a_i := cq + sa_j$ 
21:       $a_j := -sq + ca_j$ 
22:      Effect of Jacobi rotation
23:       $q := v_i$ 
24:       $v_i := cq + sv_j$ 
25:       $v_j := -sq + cv_j$ 
26:    end if
27:  end for
28: until  $maxt > tol$ 
29: for  $i = 1$  to  $n$  do
30:    $\sigma_i := \|a_i\|_2$ 
31: end for
32:  $\Sigma := \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$ 
33:  $U := A\Sigma^{-1}$ 

```

For higher accuracy, Eq. (1) is converted to adopt the fused multiply-accumulate. Additionally, $\cos(\theta)$ or $\sin(\theta)$ is corrected.

3.1 False-Position Method

We consider a real root in $f(x) = 0$. The false-position method is depicted in Fig. 1. In the initial setting, x_1 and x_2 have different values. The sign of $f(x_1)$ is set to be different from that of $f(x_2)$.

In the false-position method, x_M in Eq. (2) is set to a new position to compute the real root x in $f(x) = 0$. Accordingly, x_M is expressed as follows:

$$x_M = \frac{x_1 \times f(x_2) - x_2 \times f(x_1)}{f(x_2) - f(x_1)}. \tag{2}$$

Here, if the sign of $f(x_1)$ is the same as that of $f(x_M)$, then $x_1 \leftarrow x_M$. Alternatively, if the sign of $f(x_2)$ is the same as that of $f(x_M)$, then $x_2 \leftarrow x_M$. As depicted in Fig. 1, x_M is set to a new x_1 .

3.2 Secant Method

The secant method is depicted in Fig. 2. In this method, the following recurrence relation is adopted to compute the real root x in $f(x) = 0$:

Fig. 1 False-position method

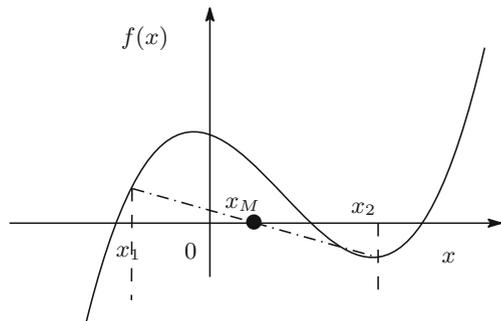
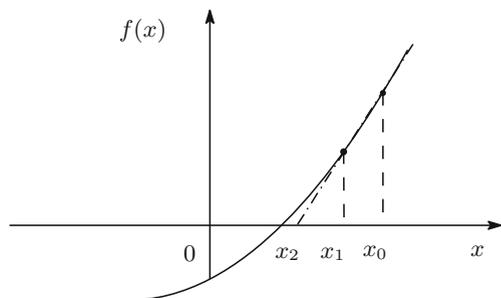


Fig. 2 Secant method



$$x_{n+1} = x_n - f(x_n) \times \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} = \frac{x_{n-1}f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})}. \tag{3}$$

From the initial setting x_0 and x_1 , the sequence of x_2, x_3, \dots converges to the real root x , as the point sequence is computed in order. When $n = 2$ in Eq. (3), we obtain Eq. (2).

3.3 Correction of $\cos(\theta)$

Computationally, $\sin(\theta)$ and $\cos(\theta)$ are affected by rounding errors. When θ is close to 0, $\sin(\theta)$ is sufficiently accurate; however, $\cos(\theta)$ often contains computation errors. In these cases, $\cos(\theta)$ is close to 1. To obtain a correct $\cos(\theta)$, we consider the following:

$$f(x) = (x)^2 + (\sin(\theta))^2 - 1 = 0. \tag{4}$$

Because $\cos(\theta)$ is close to 1, x in Eq. (4) is computed using the false-position or secant method via $x_0 = 1, x_1 = \cos(\theta)$ as follows:

$$x_2 = \frac{f(\cos(\theta)) - \cos(\theta) \times f(1)}{f(\cos(\theta)) - f(1)} = 1 - \sin(\theta) \times \frac{\sin(\theta)}{1 + \cos(\theta)} \tag{5}$$

3.4 Givens Rotation Using Fused Multiply-Accumulate

To adopt the fused multiply-accumulate, z_1 is set to

$$z_1 \leftarrow \frac{\sin(\theta)}{1 + \cos(\theta)}. \tag{6}$$

Accordingly, Eq. (1) is transformed using z_1 as follows:

$$\underline{\underline{x}} \leftarrow \sin(\theta) \left(\underline{\underline{-z_1 x + y}} \right) + x, \quad \underline{\underline{y}} \leftarrow -\sin(\theta) \left(\underline{\underline{z_1 y + x}} \right) + y. \tag{7}$$

The fused multiply-accumulate can be adopted in the double underlined part of these equations.

4 New Implementation

4.1 Improvements

Algorithm 2 shows the pseudocode of the proposed implementation. Five improvements have been made in the proposed implementation.

The first improvement is to avoid fatal bugs that could create an infinite loop. In the implementations using the Givens rotation, rounding errors may occur during orthogonalization while computing the *pair* of two vectors $(\mathbf{a}_i, \mathbf{a}_j)$. Therefore, when there is no effect of improving orthogonality in the *pair* of two vectors $(\mathbf{a}_i, \mathbf{a}_j)$, even upon performing orthogonalization, an infinite loop occurs. To avoid this infinite loop, the orthogonalization computation should be terminated when the orthogonality in all the *pair* becomes invariant.

As for the second improvement, the De Rijk method [13] is adopted. In the subroutine `jacobi_pairs` of Algorithm 1, the construction of *pair* is arbitrary and can be freely designed. The De Rijk method suggests an appropriate way to construct *pair*. As for the subroutine `jacobi_pairs`, the proposed implementation swaps the length of \mathbf{a}_i with the longest one of $\mathbf{a}_k (k = i, \dots, n)$. Thus, one can save computation cost and increase speed.

As for the third improvement, the orthogonalization computation should be minimized. In Algorithm 1, the same computation is performed for all the vectors including those that possess sufficient orthogonality. In this case, it is redundant to orthogonalize vectors that previously have sufficient orthogonality. Furthermore, the effect of rounding errors increases by repeating unnecessary orthogonalization computation. Therefore, vector \mathbf{a}_k , which is orthogonal to all the other vectors in the previous iteration, is excluded from the orthogonalization computation. This exclusion can increase accuracy and speed.

The fourth improvement is to avoid overflow and underflow. Computationally, Algorithm 1 includes the possibility of overflow and underflow while computing x and y . In the proposed implementation, lines 6 and 7 in Algorithm 1 are transformed into an equation using the length of the vector.

The fifth improvement is also to avoid overflow and underflow. Computationally, Algorithm 1 includes the possibility of overflow and underflow while computing g . To avoid this, $\mathbf{w} = \mathbf{a}_j \times t_1$ is introduced, where t_1 denotes the reciprocal of the estimated length of \mathbf{a}_j . Thus, length \mathbf{w} is almost equal to 1.

4.2 Computation with High Accuracy in $\cos(\theta)$ and $\sin(\theta)$

Jacobi rotation matrix makes the off-diagonal components zero as follows:

Algorithm 2 A pseudocode of the proposed implementation for the one-sided Jacobi method

Require: $A = [a_1 \ a_2 \ \dots \ a_n]$, $IM1 = [IM1_1, \dots, IM1_n]$, $IM2 = [IM2_1, \dots, IM2_n]$, w, tol
Ensure: (U, Σ, V)

```

1:  $V := [v_1 \ v_2 \ \dots \ v_n] := I_{n,n}$ 
2:  $\sigma_{1:n} := \|a_{1:n}\|_2$ 
3:  $IM1_{1:n} := 0$ 
4: repeat
5:    $flag := 0$ 
6:    $IM2_{1:n} := 0$ 
7:   for  $j = 1$  to  $n$  do
8:     call Algorithm 3
9:     if  $\sigma_j < SAFMIN$  then
10:       $t_1 := 1/SAFMIN$ 
11:     else
12:       $t_1 := 1/\sigma_j$ 
13:     end if
14:      $s_1 := \sigma_j \times t_1$ 
15:      $w := a_j \times t_1$ 
16:     for  $k = j + 1$  to  $n$  do
17:       call Algorithm 4
18:       if  $c \neq 1$  or  $s \neq 0$  then
19:          $IM2_j := 1$ 
20:          $IM2_k := 1$ 
21:         if  $t_4 > 0$  then
22:            $t_6 := 0$ 
23:            $t_7 := 0$ 
24:           for  $\ell = 1$  to  $m$  do
25:              $y := a_j(\ell)$ 
26:              $z := a_k(\ell)$ 
27:              $a_j(\ell) := s \times (-x \times y + z) + y$ 
28:              $w(\ell) := a_j(\ell) \times t_1$ 
29:              $t_6 := t_6 + (w(\ell))^2$ 
30:              $a_k(\ell) := -s \times (x \times z + y) + z$ 
31:              $t_7 := t_7 + (a_k(\ell) \times t_5)^2$ 
32:           end for
33:            $\sigma_j := t_3 \times \sqrt{t_6}$ ;  $s_1 := \sigma_j \times t_1$ ;  $\sigma_k := t_4 \times \sqrt{t_7}$ 
34:         else
35:            $t_6 := 0$ 
36:           for  $\ell = 1$  to  $m$  do
37:              $y := a_j(\ell)$ ;  $z := a_k(\ell)$ 
38:              $a_j(\ell) := s \times (-x \times y + z) + y$ 
39:              $w(\ell) := a_j(\ell) \times t_1$ 
40:              $t_6 := t_6 + (w(\ell))^2$ 
41:              $a_k(\ell) := -s \times (x \times z + y) + z$ 
42:           end for
43:            $\sigma_j := t_3 \times \sqrt{t_6}$ 
44:            $s_1 := \sigma_j \times t_1$ 
45:            $\sigma_k := \|a_k\|_2$ 
46:         end if
47:         if  $flag = 0$  then
48:            $t_2 := w^\top a_k$ 
49:           if  $|t_2| \leq tol \times \sigma_k \times s_1$  then
50:              $flag := 1$ 
51:           end if
52:         end if
53:          $y := v_j$ 
54:          $z := v_k$ 
55:          $v_j := s \times (-y \times x + z) + y$ 
56:          $v_k := -s \times (-z \times x + y) + z$ 
57:       end if
58:     end for
59:   end for
60:    $IM1 = IM2$ 
61: until  $flag = 1$ 
62:  $\sigma_{1:n} := \|a_{1:n}\|_2$ 
63:  $\Sigma := \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$ 
64:  $U := A\Sigma^{-1}$ 

```

Algorithm 3 A pseudocode of the proposed implementation for the swap part of the one-sided Jacobi method

```

1: if  $IM1_j = 0$  or  $\sigma_j = 0$  then
2:   cycle
3: end if
4:  $i := j$ 
5: for  $\ell = j + 1$  to  $n$  do
6:   if  $IM1_\ell = 1$  and  $\sigma_\ell > \sigma_i$  then
7:      $i := \ell$ 
8:   end if
9: end for
10: if  $i \neq j$  then
11:   swap  $a_i, a_j$ 
12:   swap  $v_i, v_j$ 
13:   swap  $\sigma_i, \sigma_j$ 
14:   swap  $IM1_i, IM1_j$ 
15:   swap  $IM2_i, IM2_j$ 
16: end if
17: if  $IM1_j = 0$  or  $\sigma_j = 0$  then
18:   cycle
19: end if

```

$$\begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} \mathbf{a}_j^\top \mathbf{a}_j & \mathbf{a}_j^\top \mathbf{a}_k \\ \mathbf{a}_j^\top \mathbf{a}_k & \mathbf{a}_k^\top \mathbf{a}_k \end{pmatrix} \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} = \begin{pmatrix} \hat{\beta}_{j,j} & 0 \\ 0 & \hat{\beta}_{k,k} \end{pmatrix}$$

Generally, $\cos(\theta)$ and $\sin(\theta)$ in Eq. (8) are computed as follows:

$$f = \frac{(\mathbf{a}_j^\top \mathbf{a}_j - \mathbf{a}_k^\top \mathbf{a}_k)}{2}, \quad g = \mathbf{a}_j^\top \mathbf{a}_k, \quad t = \frac{g}{\left(f + \text{sign}\left(\sqrt{g^2 + f^2}, f\right)\right)} \quad (8)$$

$$r = \sqrt{1 + t^2}, \quad \cos(\theta) = \frac{1}{r}, \quad \sin(\theta) = \frac{t}{r}, \quad (9)$$

However, in the proposed implementation, the following equations are adopted to compute f and g :

$$f = \frac{\left(\sqrt{\mathbf{a}_j^\top \mathbf{a}_j} - \sqrt{\mathbf{a}_k^\top \mathbf{a}_k}\right) \times \left(\sqrt{\mathbf{a}_j^\top \mathbf{a}_j} \times t_1 + \sqrt{\mathbf{a}_k^\top \mathbf{a}_k} \times t_1\right)}{2}, \quad (10)$$

$$g = \left(\mathbf{a}_j^\top \mathbf{a}_k\right) \times t_1 = \mathbf{w}^\top \mathbf{a}_k. \quad (11)$$

The proposed implementation can avoid divergence using w .

Algorithm 4 A pseudocode of the proposed implementation for computing the Jacobi rotation of the one-sided Jacobi method

```

1: if  $\sigma_j = 0$  then
2:   exit
3: end if
4: if  $IM1_k = 0$  or  $\sigma_k = 0$  then
5:   cycle
6: end if
7:  $t_2 := \mathbf{w}^\top \mathbf{a}_k$ 
8: if  $|t_2| \leq tol \times \sigma_k \times s_1$  then
9:   cycle
10: end if
11:  $s_2 := \sigma_k \times t_1$ 
12:  $f := (\sigma_j - \sigma_k) \times (s_1 + s_2)/2$ 
13:  $g := t_2$ 
14:  $t := g / \left( f + \text{sign} \left( \sqrt{g^2 + f^2}, f \right) \right)$ 
15:  $r := \sqrt{1 + t^2}$ 
16:  $c = 1/r$ 
17:  $s = t/r$ 
18:  $t_3 := \sigma_j \times \sqrt{t \times (t_2 / (\sigma_j \times s_1)) + 1}$ 
19: if  $t_3 < SAFMIN$  then
20:    $t_3 := SAFMIN$ 
21: end if
22:  $t_1 := 1/t_3$ 
23:  $t_8 := -t \times (t_2 / (\sigma_k \times s_2)) + 1$ 
24: if  $t_8 > 0$  then
25:    $t_4 := \sigma_k \times \sqrt{t_8}$ 
26:   if  $t_4 < SAFMIN$  then
27:      $t_4 := SAFMIN$ 
28:   end if
29:    $t_5 := 1/t_4$ 
30: else
31:    $t_4 := 0$ 
32: end if
33:  $x := s/(1 + c); c := -s \times x + 1$ 

```

5 Experiments

We evaluated the proposed implementation to see if it had higher accuracy than those of the QR method, OQDS method in [1], and xGESVJ implemented in LAPACK-3.9.0, which is a routine for the one-sided Jacobi method. Because the test matrices are upper triangular, the Householder transformation is adopted as a preprocessing method for the QR and OQDS methods.

5.1 Environment

Table 1 summarizes the experimental environment. We use four real random upper triangular matrices, whose dimensions are 500×500 , 1000×1000 , 1500×1500 , and 2000×2000 , respectively. The following Frobenius norms are used to evaluate the computation errors:

$$\|A - U\Sigma V^T\|_F, \quad \|U^T U - I\|_F, \quad \|V^T V - I\|_F. \quad (12)$$

5.2 Result and Consideration

Figure 3 depicts the performance results. From Fig. 3a, it is evident that the proposed implementation requires approximately 1.3 times the computation time of `xGESVJ`. It is caused that computation time in the Givens rotation with high accuracy is higher than that in the Jacobi rotation in `xGESVJ`. From Fig. 3b, c, it is evident that $\|A - U\Sigma V^T\|_F$ and $\|U^T U - I\|_F$ in the proposed implementation are the highest among all the implementations. It is caused that the QR and OQDS methods are affected by rounding errors due to preprocessing via the Householder transformation; the Givens rotation of the proposed implementation with high accuracy is better than that of `xGESVJ`. Especially, from Fig. 3c, we can see that the orthogonality of left singular vectors in the proposed implementation is approximately ten times more accurate than that in `xGESVJ`. Table 2 shows the comparison in $\|V^T V - I\|_F$. From Fig. 3d, we say that $\|V^T V - I\|_F$ in the proposed implementation is smaller than that in the QR method and `xGESVJ`. From Table 2, $\|V^T V - I\|_F$ in the proposed implementation is slightly larger than that in the OQDS method but not considerably different.

In this study, we aimed to implement a highly accurate one-sided Jacobi method. Experimental results confirm that although the computation time of the proposed implementation is approximately 1.3 times that of the conventional implementation, its accuracy is significantly higher. In the Sakurai–Sugiura method, only left singular vectors U are required. Therefore, the orthogonality of right singular vector V is not

Table 1 Experimental environment

CPU	Intel(R) Core(TM) i3-6100 CPU 3.70GHz
OS	Fedora 32
RAM	16GB
Cache	3MB
Compiler	gfortran 10.1.1
Options	-O3 -mtune=native -march=native
Software	Lapack-3.9.0
Precision	Single precision

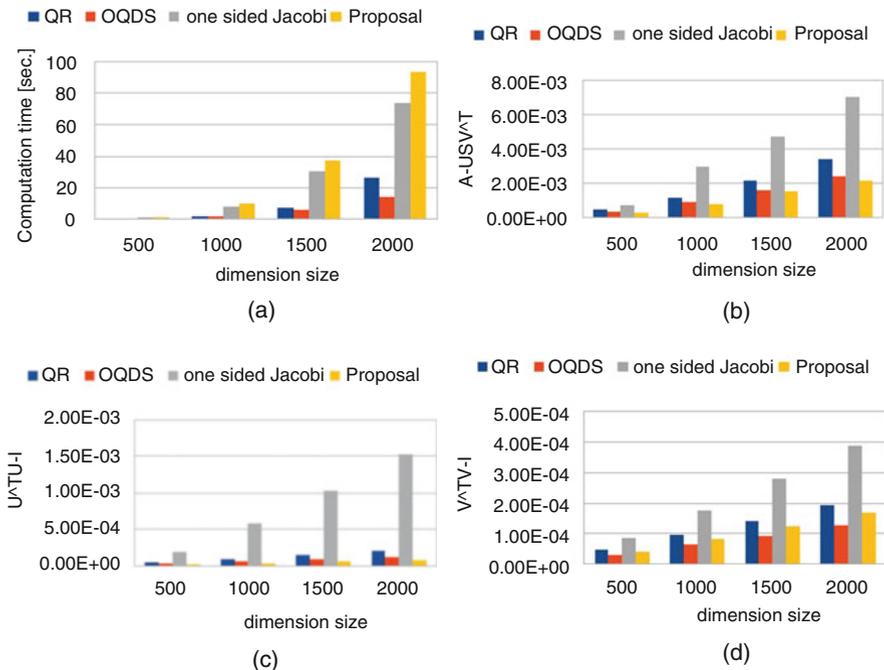


Fig. 3 Comparison. (a) Computation time (s) (b) $\|A - U\Sigma V^T\|_F$ (c) $\|U^T U - I\|_F$ (d) $\|V^T V - I\|_F$

Table 2 Comparison in $\|V^T V - I\|_F$. (10^{-5})

Dimension size	QR	OQDS	One-sided Jacobi	Proposal
500	4.86	3.12	8.20	4.15
1000	9.73	6.33	17.82	8.33
1500	14.16	9.43	28.11	12.49
2000	19.09	12.62	39.14	16.69

relevant. Furthermore, $\|V^T V - I\|_F$ in the proposed implementation is comparable to that in the OQDS method. Hence, the proposed implementation of the one-sided Jacobi method with high accuracy is appropriate for the Sakurai–Sugiura method.

6 Conclusion

We proposed a novel implementation of the one-sided Jacobi method with high accuracy. In the implementation, a Givens rotation with high accuracy and the fused multiply-accumulate were adopted. Notably, overflow and underflow may occur in the conventional method [4]. To avoid this problem, we have carefully improved

the one-sided Jacobi method. Because the Givens rotation with high accuracy requires considerable computation time, we reduced redundant computation to speed up the operation. The experimental result confirmed that the proposed implementation required approximately 1.3 times the computation time as that of routine in LAPACK. Furthermore, the orthogonality of the left singular vectors in the proposed implementation was the best among all the implementations. In the Sakurai–Sugiura method, only left singular vectors are required. Consequently, the proposed implementation realized the one-sided Jacobi method with high accuracy. Hence, our proposed method might be applied to the Sakurai–Sugiura method.

As a future work, we wish to further speed up the proposed method.

Acknowledgments This work was supported by JSPS KAKENHI Grant Numbers JP17H02858 and JP17K00167.

References

1. S. Araki, H. Tanaka, M. Takata, K. Kimura, Y. Nakamura, Fast computation method of column space by using the DQDS method and the OQDS method, in *Proceedings of PDPTA 2018* (2018), pp. 333–339
2. R.P. Brent, F.T. Luk, C. van Loan, Computation of the singular value decomposition using mesh-connected processors. *J. VLSI Comput. Syst.* **1**, 242–270 (1985)
3. J. Demmel, *Applied Numerical Linear Algebra* (SIAM, Philadelphia, 1997)
4. J. Demmel, K. Veselic, Jacobi's method is more accurate than QR. *SIAM J. Matrix Anal. Appl.* **13**(4), 1204–1245 (1992)
5. J. Drmac, K. Veselic, New fast and accurate Jacobi SVD algorithm: I. *SIAM J. Matrix Anal. Appl.* **29**, 1322–1342 (2008)
6. Z. Drmac, K. Veselic, New fast and accurate Jacobi SVD algorithm: II. *SIAM J. Matrix Anal. Appl.* **29**, 1343–1362 (2008)
7. K.V. Fernando, B.N. Parlett, Accurate singular values and differential qd algorithms. *Numer. Math.* **67**, 191–229 (1994)
8. G.E. Forsythe, P. Henrici, The cyclic Jacobi method for computing the principal values of a complex matrix. *Trans. Am. Math. Soc.* **94**, 1–23 (1960)
9. E. Kogbetliantz, Solution of linear equations by diagonalization of coefficients matrix. *Q. Appl. Math.* **13**, 123–132 (1955)
10. Linear Algebra PACKage. <http://www.netlib.org/lapack/>. Last Accessed 10 Mar 2020
11. U. von Matt, The orthogonal qd-algorithm. *SIAM J. Sci. Comput.* **18**, 1163–1186 (1997)
12. B.N. Parlett, O.A. Marques, An implementation of the dqds algorithm (positive case). *Lin. Alg. Appl.* **309**(1–3), 217–259 (2000)
13. P.P.M. De Rijk, A one-sided Jacobi algorithm for computing the singular value decomposition on a vector computer. *SIAM J. Sci. Stat. Comput.* **10**, 359–371 (1998)
14. H. Rutishauser, The Jacobi method for real symmetric matrices. *Numer. Math.* **9**(1), 1–10 (1966)
15. T. Sakurai, H. Tadano, CIRR: A Rayleigh-Ritz type method with counter integral for generalized eigenvalue problems. *Hokkaido Math. J.* **36**, 745–757 (2007)

Improvement of Island Genetic Algorithm Using Multiple Fitness Functions



Shigeka Nakajima and Masami Takata

1 Introduction

In the island genetic algorithm (GA), individuals in each island evolve. In the natural world, environmental differences in habitats such as the sky, sea, and land differ significantly. Even in the same land area, different forms of living organisms exist in dry zones and wetlands. Each should evolve in a manner that suits the environment, and owing to the various environments on Earth, various organisms have been diversified. In addition, individuals in distant areas do not interact with each other and do not exchange genes; therefore, they evolve uniquely in that area.

The island GA is a method for solving an optimization problem using diversity. In the island GA, each island generates offspring from the parent. In a generation, multiple individuals migrate to different islands. By repeating this operation, offspring generation evolves with diversity.

In the conventional island GA, offspring generation evolves based on the same fitness function. Meanwhile, in the natural world, different islands have different fitness functions. Therefore, in this study, we improve the setting for fitness functions on each island. Hence, each island has a distinctive evolution.

In Sect. 2, we introduce a packing problem, which is a multi-objective optimization problem. In Sect. 3, we explain the island GA with different fitness functions. In Sect. 4, we evaluate the improved island GA.

S. Nakajima · M. Takata (✉)
Nara Women's University, Nara, Japan
e-mail: takata@ics.nara-wu.ac.jp

2 Packing Problem

Packing problems are known as NP-hard problems. To solve these problems, the bottom-left (BL) [2], best-fit [1], and BDL methods [4] have been proposed.

In the BL method, the packing order of objects is determined, and the packing operation is minimized. If the height is the same, then the operation of packing to the left is repeated as much as possible. In the best-fit method, the operation of packing objects that allows the maximum usage of the gap is repeated. The BL and best-fit methods are for a two-dimensional space, whereas the BDL method is intended for a three-dimensional space.

In the BDL method, all objects are assigned random numbers. Objects are arranged at the bottom, depth, and left in numerical order. Subsequently, by changing the order of the placed objects, the fitness of the BDL method is improved. In each iteration, the objects are packed in the following priority order: the bottom, deepest, and leftmost.

First, the objects are left aligned and placed at the left edge. Next, when it arrives to the right position, the placement start position is updated to the front, and the left alignment is performed again. After deciding the placement of the lower tier, the same procedure is performed to place it on the upper tier.

Let i be the number for packing the object. Let (x_i, y_i, z_i) be the vertex closest to the origin. d_i , w_i , and h_i represent the depth, width, and height of the object, respectively. When the i -th object is placed, $(x_i + d_i, y_i, z_i)$, $(x_i, y_i + w_i, z_i)$, and $(x_i, y_i, z_i + h_i)$ become the candidates in the $(i + 1)$ -th and subsequent objects instead of (x_i, y_i, z_i) .

Figure 1 shows the arrangement of items i and $(i + 1)$. The red dot represents a candidate for the starting point. In the BDL method, the arrangement order is uniquely determined.

In the BDL method, fitness functions f_1 , f_2 , and f_3 represent the frequency of use, capacity, and viewability, respectively. They are combined to evaluate the total fitness function $p(f_1, f_2, f_3)$. Using the combination fitness functions, the objects are

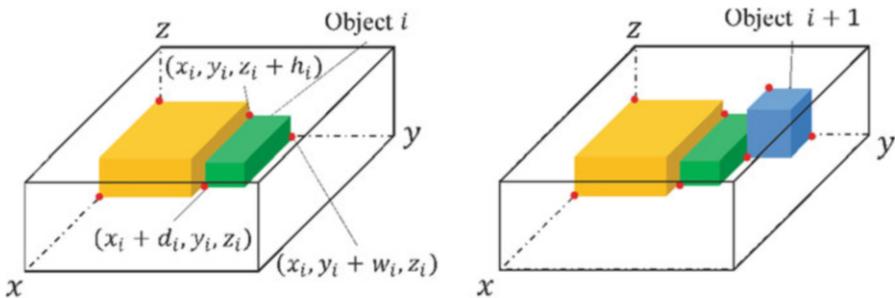


Fig. 1 Arrangement of i and $i + 1$

compared when they are swapped and not swapped. Therefore, when many objects exist, the package problem is NP-hard.

Objects have a usage frequency. Objects with a high usage frequency should be placed at a location that facilitates placement and retrieval. In other words, a frequently used object should be placed in front of the drawer to ease its retrieval, and an infrequently object should be placed behind the drawer. The effort to retrieve an item is defined as the cost. The object that overlaps in the back of the drawer is considered as lower in cost, whereas that at the top is lower in cost. If u_i is the frequency of use, and k is the number of objects on the top, the cost \tilde{u}_i is

$$\tilde{u}_i = \frac{u_i \left(D - \left(x_i + \frac{d_i}{2} \right) \right)}{D} + \sum_k \tilde{u}_{i_k}. \tag{1}$$

The total cost \tilde{u}_i should be small. Therefore,

$$f_1 = - \sum_i^b \tilde{u}_i. \tag{2}$$

When packing objects, it is preferable to arrange it such that it is easily viewable from above. When two objects are piled up and put away, if one object is completely hidden by an object above it, it is impossible to confirm what is underneath and the object on top must be removed. Therefore, an optimal arrangement is required such that more objects are visible and can be distinguished when viewed from above. Let o_i be the area visible from the top of each object, and the ratio r_i visible from the top is

$$r_i = \frac{o_i}{d_i w_i}. \tag{3}$$

Let R be the lower limit that allows r_i and the score l_i to determine whether each item can be viewed from the above be expressed as follows:

$$l_i = 1 (r_i > R) \quad l_i = 0 (r_i < R) \tag{4}$$

Using l_i , the visibility f_2 from the top of the entire drawer is

$$f_2 = \sum_i^b l_i. \tag{5}$$

The evaluation of the capacity, f_3 , is expressed by the number of objects contained. In other words, a larger value is preferred.

Hence, the evaluation function of the BDL method is expressed as

$$p (f_1, f_2, f_3) = f_1 + f_2 + f_3. \tag{6}$$

3 Island GA with Different Fitness Functions

3.1 Concept

Let $S_i (i = 1, \dots, n)$ be an island, and let $f_j (j = 1, \dots, m, m \leq n)$ be an objective function. In a GA for multiple objectives [3], all objective functions are combined into a single fitness function $p(f_1, f_2, f_3)$. Subsequently, generational change is performed using the same fitness function. In this study, the evaluation function of each island was used as the fitness function unique to each island because the purpose was to create a distinctive evolution. In other words, the fitness function of the island S_i is f_j .

The fitness function of $S_k (k = m + 1, \dots, n)$ should be set appropriately using the function represented by f_j or its combination.

The procedure to create the proposed island GA with different fitness functions is as follows:

1. Set up island S_i .
2. Generate an initial individual D_j on each island S_i .
3. Perform crossover on the same island.
4. Apply mutations to the individuals.
5. Calculate the fitness value of each individual using the fitness function f_i for each island S_i .
6. Select offspring generation.
7. Repeat steps 4–8 for g_1 generations.
8. Select t individuals to migrate from island S_i .
9. Immigrate.
10. Calculate the fitness value of the migrating individual using the fitness function f_i for each island S_i .
11. If the number of generations becomes the last generation g_2 , the algorithm is terminated.
12. Return to step 4.

In step 1, islands are defined. In this proposal, when the multipurpose objective function is m , the number of islands is m or higher. A function is used only on one island and not on the other. When combination functions are defined, islands are prepared. Therefore, the number of islands must be larger than m .

In step 2, the initial individual D_i is generated on each island. Each individual has a gene length of l .

In step 3, new individuals are generated using the crossover.

In step 4, the mutation is adapted to the individual since it can prevent a local solution with a biased evaluation value. Some of the genes of the individual are changed randomly.

In step 5, the evaluation value of each individual on each island is calculated. Our goal is that the islands achieve a distinctive evolution through the fitness function of each island $S_i f_j$. In the conventional island GA, the function $p(f_1, \dots, f_m)$ is used on

all islands. However, it is difficult to set the appropriate parameter $p(f_1, \dots, f_m)$. Therefore, because the parameters of $p(f_1, \dots, f_m)$ are often set experimentally, a fitness function f_i can interfere with the optimization of the others. Hence, Galapagos individuals with higher f_i can be generated to avoid this problem.

In step 6, offspring are selected. At this point, selection methods have been proposed, such as elite, roulette, tournament, and random.

In step 7, steps 4–7 are repeated for g_1 generations.

In step 8, individuals are selected to migrate from the island. When t is extremely small, the effect of immigration cannot be obtained. When t is large, the ecosystem of the distinctive evolution may be damaged.

In step 9, immigration is performed. An immigrant island is randomly selected.

In step 10, the evaluation value of the immigrated individual is calculated using the objective function f_i for each island S_i .

Step 11 is the termination condition.

3.2 Application to Packing Problem Using BDL Method

To confirm the effect of the proposed island GA, we adopted the packing problem using the BDL method.

In this study, objects $M_i (i = 1, \dots, b)$ were prepared. The size of each object was set randomly. The length of one side should not exceed the size of a case. The usage frequency is randomly assigned to each object in five levels. The smaller the level, the higher is the usage frequency. The BDL method has three objective functions, as explained in Sect. 2. f_1 is the frequency of use, and f_2 is the visibility. In this study, f_3 is the total volume of the objects, not the number of objects. Subsequently, the total fitness function is that shown in Eq. (6). Because Eq. (6) has not been validated, only f_i was used as the fitness function for each island in this study. Therefore, the number of islands was three.

Let N be the number of individuals on each island. In an individual $D_i (i = 1, \dots, N)$, the order of the objects in the case is provided as the gene information. We explain this according to the procedure described in Sect. 3.1.

In step 1, three islands are generated.

In step 2, individuals D_i are generated on each island, and a random number is set as a gene. Because the same object is not selected multiple times, different numbers are assigned as the gene. In this study, the n best individuals of the parent generation are selected as offspring generation using an elite selection.

In step 3, $N - n$ individuals are generated by operating a crossover on the same island. In this section, the order of the objects is optimized. Therefore, a partially matched crossover (PMC) [1] is adopted. In the PMC, the following procedure is performed for the individual's parent₁ and parent₂:

- I. Select two crossover points.

- II. Swap the gene information between the crossover points with parent₁ and parent₂.
- III. In the swap, a series of mappings is defined.
- IV. Fill in additional genes from the original parents, for which no conflict exists.

Figure 2 shows an example of the procedure. The thin arrow corresponds to step III above. In other words, in procedure III, $1 \leftrightarrow 3$ and $4 \leftrightarrow 5$ are obtained. Here, “6” is included in both swapped intervals; therefore, $1 \leftrightarrow 6$ and $3 \leftrightarrow 6$ are deleted from the mapping.

In step 4, the probability of mutation is set to 5%. If the probability is extremely low, it tends to fall into a local solution; however, if it is extremely high, the evaluation value tends to converge.

In step 5, the fitness value of the individuals on island S_i is computed using f_i . If the objects in an individual should overflow, then the fitness value is computed using only the incoming objects.

In step 6, through the remaining n individuals from the parental generation by the elite selection, the individuals are added to the offspring generation. Therefore, the elite information can be retained while discarding diversity.

In step 8, two immigrants are randomly selected.

Figure 3 shows a conceptual diagram of step 9.

Fig. 2 Crossover

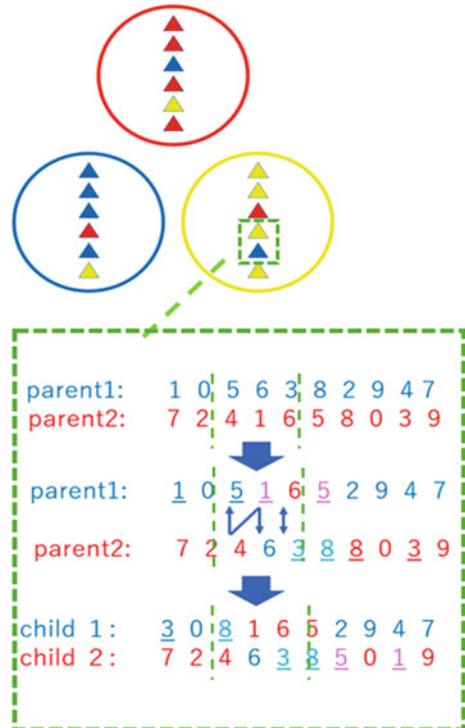


Fig. 3 Selection

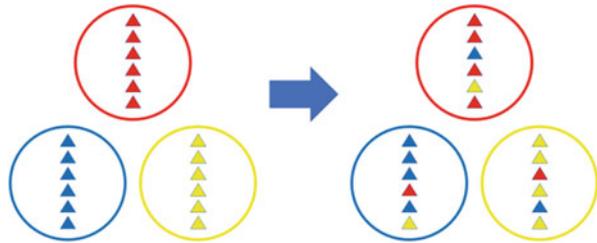


Table 1 Parameters

Size of object	Height: 1–30Width: 1–30Depth: 1–30
Frequency of use	1–5
Number of objects	100
Mutation rate	5%
Crossover rate	80%
Crossover method	Partially matched crossover
Selection method	Elite selection

4 Experiment

4.1 Experimental Setting

In the proposed island GA, we improved different fitness values for each island. Therefore, each island can proceed independently. In this section, we compare the conventional and the proposed island GAs. In the conventional island GA, all islands evolve using $p(f_1, f_2, f_3)$. Meanwhile, in the proposed island GA, each island evolves using the difference f_i .

Table 1 shows the parameters used in the experiments. The size and usage frequency of the objects were set randomly within the range shown in Table 1. The number of boxes stored was the gene length.

4.2 Experimental Result

Figure 4 shows the best fitness generation in the conventional island GA. In Fig. 4, for the conventional island GA, it can be confirmed that all f_i evolved with the evolution of $p(f_1, f_2, f_3)$. Furthermore, f_3 and $p(f_1, f_2, f_3)$ showed the same tendency. Meanwhile, f_1 and f_2 were almost flat. Therefore, it was assumed that f_1 and f_2 did not significantly affect the conventional GA.

Figure 5 shows the best fitness generation in the proposed island GA. The orange, green, and blue lines express the results on islands S_1 , S_2 , and S_3 , respectively. The left and right graphs show the generation in $p(f_1, f_2, f_3)$ and f_i , respectively. As shown from the right graphs in Fig. 5, it is confirmed that the best fitness value increased as the island was generated. In the proposed island GA, it is confirmed that f_3 evolved

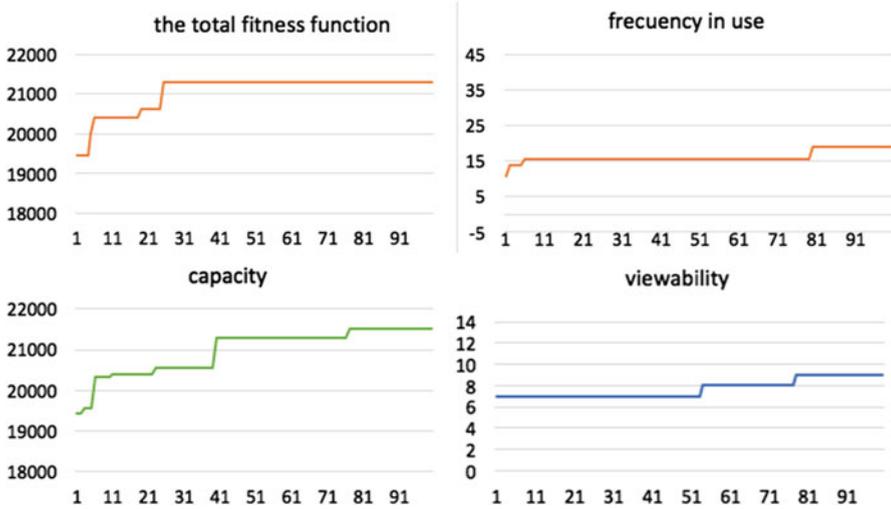


Fig. 4 Best fitness generation in the conventional island GA

with the evolution of $p(f_1, f_2, f_3)$. Meanwhile, in islands S_1 and S_2 , f_i increased as the generation progressed, but $p(f_1, f_2, f_3)$ indicated a zigzag pattern.

As shown in Figs. 4 and 5, all f_i of the proposed island GA were higher than that in the conventional island GA. This is attributable to the synergistic effect of producing excellent individuals for each f_i by distinctive evolution. In the proposed island GA, although the change in f_3 was small, f_1 and f_2 indicated better fitness values. Therefore, the proposed island GA was effective.

Figures 6 and 7 show the distribution diagrams of the initial and final generations g_2 in the conventional island GA and the proposed island GA, respectively. In Figs. 6 and 7, ai , bi , and ci show the results for islands S_1 , S_2 , and S_3 , respectively. Meanwhile, $a1$, $a4$, $b1$, $b4$, $c1$, $c2$ in Figs. 6 and 7 are the distributions in f_1 and f_2 ; $a2$, $a5$, $b2$, $b5$, $c2$, $c5$ are the distributions in f_1 and f_3 ; and $a3$, $a6$, $b3$, $b6$, $c3$, $c6$ are the distributions in f_2 and f_3 . As shown in Figs. 6 and 7, the proposed GA has a greater individual dispersion in the final generation than the conventional island GA, and diversity is secured. This indicates that the distinctive evolution in the proposed island GA is effective. Hence, the proposed island GA can generate better individuals.

Because the proposed island GA maintains its diversity by rendering it a distinctive evolution, it is unlikely to fall into a local solution. Therefore, it may yield better results than the conventional island GA, rendering it an effective solution.

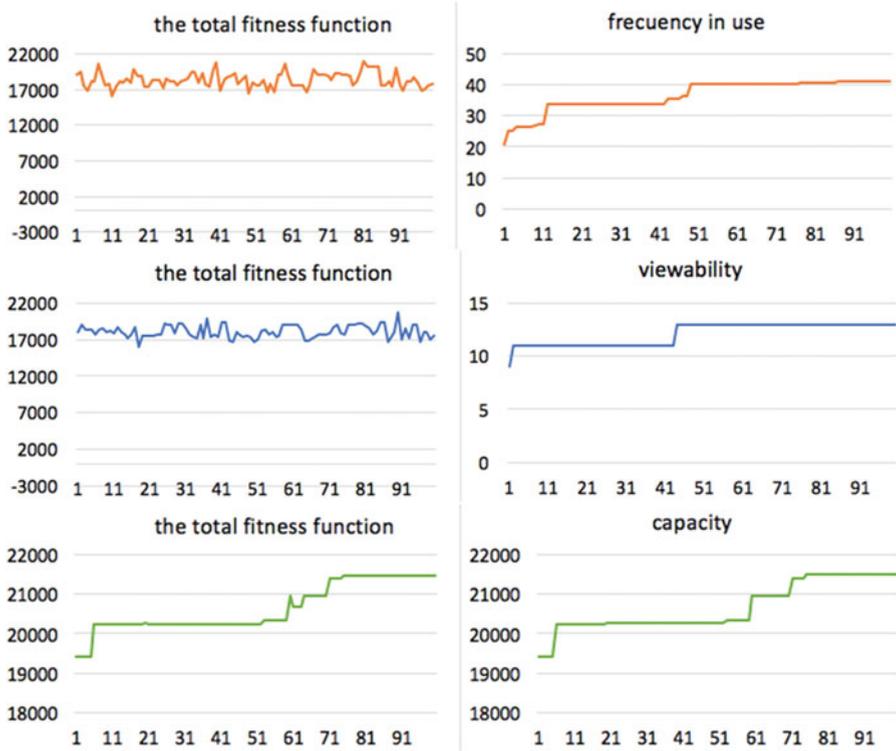


Fig. 5 Best fitness generation in the proposed island GA

5 Conclusion

In this study, we improved the conventional island GA, which uses the same fitness function. In the proposed island GA, each island aims at different fitness functions. The experimental results indicated that using the proposed island GA, the diversity can be maintained, and the local solution can be appropriately prevented. Hence, distinctive evolution performed well in the improved island GA.

As a future study, we plan to apply the proposed island GA to an optimization problem, in which the correlation plane of a multi-objective objective function is represented by a concave plane.

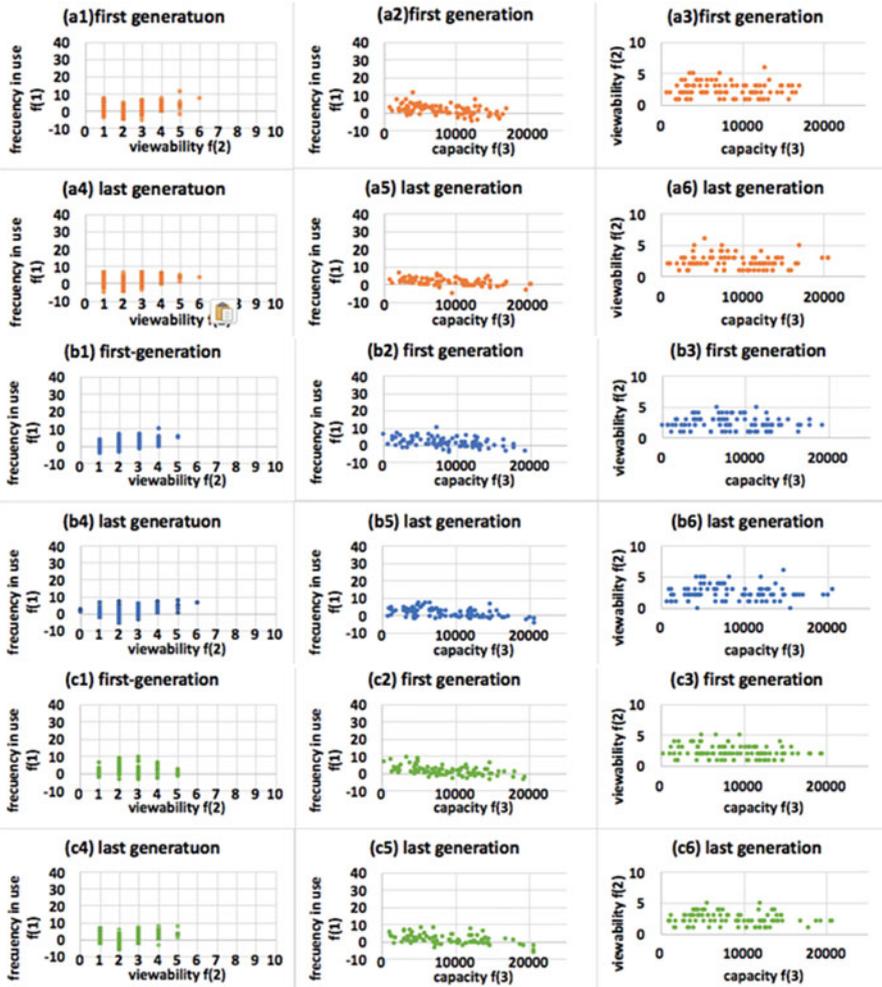


Fig. 6 Distribution diagrams of the initial and final generations in the conventional island GA

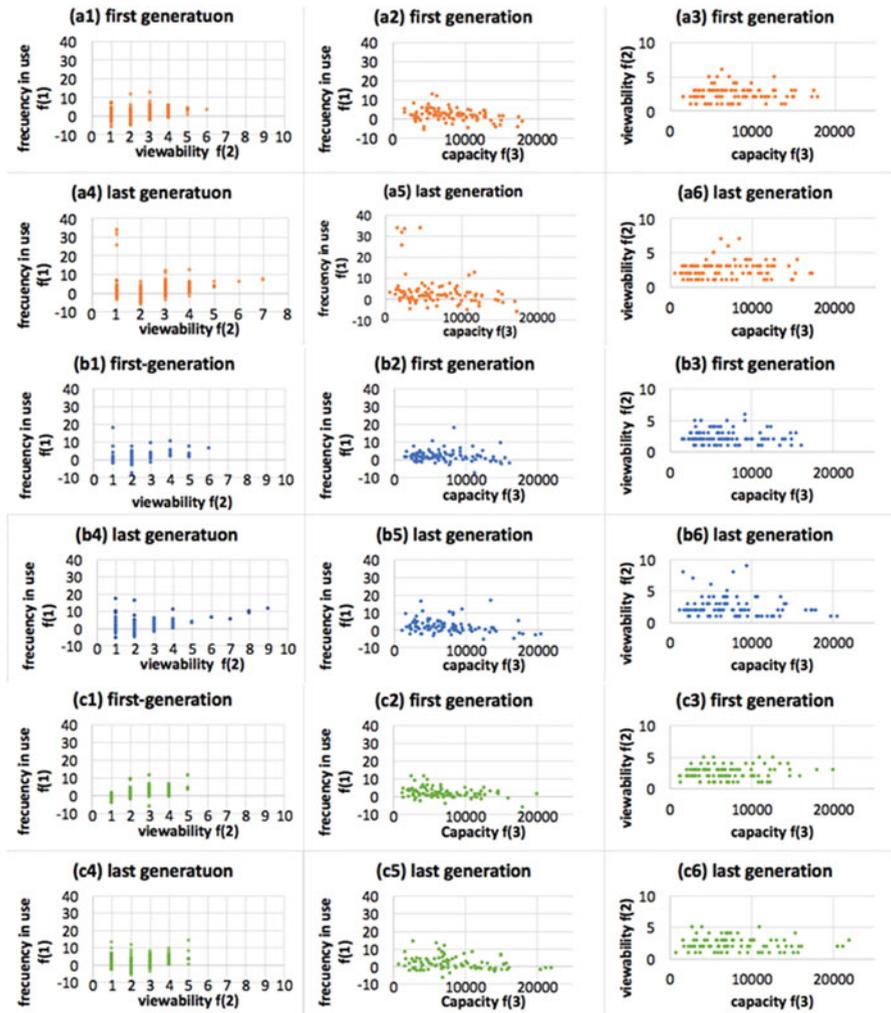


Fig. 7 Distribution diagrams of the initial and final generations in the proposed island GA

Acknowledgments We would like to thank Editage (www.editage.com) for English language editing.

References

1. B.S. Baker, E.G. Coffman, R.L. Rivest Jr., Orthogonal packing in two dimensions. *SIAM J. Comput.* **9**, 846–855 (1980)
2. T. Fujita, Y. Numata, T. Yoshimi, A study of intelligent drawer with RFID tag information reading system for intelligent space, in *Proceedings of 2012 IEEE International Conference on Mechatronics and Automation* (2012), pp. 533–538

3. D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, Reading, 1989)
4. N. Yano, T. Morinaga, T. Saito, Packing optimization for cargo containers. SICE Annu. Conf. **2008**, 3479–3482 (2008)

High-Performance Cloud Computing for Exhaustive Protein–Protein Docking



Masahito Ohue, Kento Aoyama, and Yutaka Akiyama

1 Introduction

The cloud computing environment is regarded as an important computing resource in large-scale data analysis [1, 2]. It is often used for calculation and analysis accompanied by big data, such as genomics and biomedicine [3, 4]. The development of public clouds such as Microsoft Azure, Amazon Web Services, and the Google Cloud Platform has contributed to the performance of large-scale bioinformatics analysis on the cloud environment [5–7]. Bioinformatics problems including sequence homology searches (BLAST and others) [8–10], similarity searches of tertiary protein structures [11, 12], ab initio tertiary protein structure prediction [13], quantitative structure–activity relationship modeling [14], and

M. Ohue (✉)

Department of Computer Science, School of Computing, Tokyo Institute of Technology, Yokohama City, Kanagawa, Japan

Ahead Biocomputing, Co. Ltd., Kawasaki City, Kanagawa, Japan

e-mail: ohue@c.titech.ac.jp; ohue@ahead-biocomputing.co.jp

K. Aoyama

Department of Computer Science, School of Computing, Tokyo Institute of Technology, Tokyo, Japan

RWBC-OIL, National Institute of Advanced Industrial Science and Technology, Tsukuba, Ibaraki, Japan

e-mail: aoyama@bi.c.titech.ac.jp

Y. Akiyama

Department of Computer Science, School of Computing, Tokyo Institute of Technology, Tokyo, Japan

Ahead Biocomputing, Co. Ltd., Kawasaki City, Kanagawa, Japan

e-mail: akiyama@c.titech.ac.jp; akiyama@ahead-biocomputing.co.jp

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_53

737

protein–ligand docking [15, 16] are applied in cloud computing environments as a computing resource.

Among the numerous merits of several existing cloud computing platforms, the pay-as-you-go concept whereby a user can use as much as he/she wishes at any time is the greatest advantage. Large-scale parallel computing using supercomputers enables large-scale simulation and processing of substantial amounts of data, but a user account approval procedure is required according to the institutional rules, or the services are available only for the member of the organization possessing the supercomputer. In particular, several barriers exist to use for commercial purposes and owing to factors such as publicness, security, and national strategy in supercomputer at public institution. Generally it is difficult for external people to use the public institution supercomputer casually. However, if it is on a cloud, anyone can use computational resources on thousands of cores instantly when necessary.

Distributed computing has mainly been selected as the method for cloud computing. With the development of grid computing, computation on the cloud by Apache Hadoop has been conducted extensively [3, 5, 7, 8], and support tools for constructing Hadoop clusters on the cloud have been established [17]. However, while Hadoop/MapReduce can easily construct a distributed task calculation environment, it is versatile and therefore contains an excessive amount of functions. These tools exhibit limited applicability to certain areas such as data mining, because MapReduce provides poor performance on problems with an iterative structure present in the linear algebra that underlies a substantial amount of data analysis [18]. To improve the performance and enable flexible design according to scientific applications, an original task distribution system has been constructed based on the message passing interface (MPI) in several cases [9]. Hassan et al., for example, implemented well-known MPI-based benchmarks (NAS parallel benchmarks) in Azure [19].

Fortunately, AWS and Azure provide instances and networks with awareness of parallel high-performance computing (HPC). For example, in Azure, which was used in this research, an instance of a remote direct memory access (RDMA) network (InfiniBand) is also provided. Such an environment is expected to highly effect for parallel computing applications. However, information such as which instance should be used, the amount of scalability obtained, and the price has not been sufficiently clarified in previous studies.

Therefore, in this study, a large-scale parallel computation of a bioinformatics application was performed on several cloud instances with suggestions for the choice of the public cloud usage environment. We focused on protein–protein interaction predictions, particularly the protein–protein docking problem, as a bioinformatics application. Protein–protein docking, which is a computational method for predicting the structure of a protein complex from known component structures, is a powerful approach that facilitates the discovery of otherwise unattainable protein complex structures. Fast Fourier transform (FFT)-based rigid-body initial protein–protein docking tools are the mainstream of protein–protein docking (as reviewed by Matsuzaki et al. [20]). Several applications also require a huge number of dockings, such as consensus-based refinement [21, 22], large-

scale interactome predictions [23–25], the identification of protein binders [26, 27], and multiple docking [28]. We previously developed the supercomputer-powered software MEGADOCK [24, 29, 30], and we drew on this experience to develop a protein–protein docking tool for efficient HPC computation on the public cloud. A protein–protein docking environment that can achieve large-scale analysis on the cloud is necessary in the current global situation, in which large-scale computing environments are readily available on the cloud.

In this study, we demonstrated the implementation and performance of high-performance cloud protein–protein docking. We evaluated the parallelization efficiency (strong scaling) of MEGADOCK implemented on Microsoft Azure and verified its usage efficiency for GPU instances.

2 Materials and Methods

2.1 Configuration of Azure Cloud Computing Environment

A unit of computing environment on Azure is called an instance or virtual machine (VM). The machine architecture on Azure is composed of multiple VMs and storage, as illustrated in Fig. 1. Each VM and storage is first deployed from Azure CLI and then registered as a resource group in Azure. Thereafter, the computation task is executed on multiple VMs by means of MPI communication. The programs for the bulk VM deployment and bulk undeployment were developed in this study.

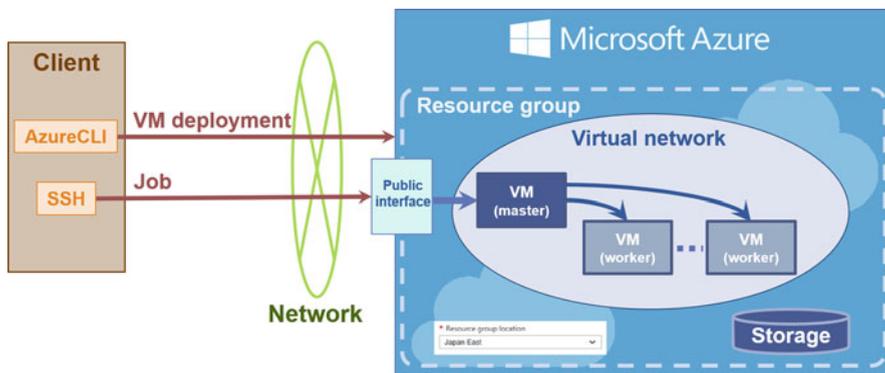


Fig. 1 Configuration of Azure cloud computing environment

2.2 MEGADOCK: Protein–Protein Docking Tool

MEGADOCK [30] is our software for protein–protein interaction prediction. The 3D structures (PDB data) of two proteins for predicting interaction are input, and the presence or absence of the interaction is output in the form of a score. The main part of the calculation is grid-based docking of the protein, which is implemented using FFT [31]. The FFT calculation depends on the protein size but is approximately 80% of the total occupancy. The computational scale is $O(N^3 \log N)$ if the size of one side of the grid is N , usually representing a protein in a grid of 1.2 Å pitches.

MEGADOCK is a multi-threaded implementation that uses OpenMP and runs on a multi-core CPU. Furthermore, a GPU-implemented version is available, which runs on the multiple GPUs using the CUDA library [32]. A multi-node parallel implementation version was also created by hybrid parallelization combined with MPI parallelization [24]. In this work, we constructed parallel implementations for both the CPU VMs and GPU VMs. The details of the parallelization are presented in the following subsection.

2.3 Handling Multiple VMs

In the multi-node implementation of MEGADOCK, a master–worker-type task dispatching is performed using MPI. Specifically, one process becomes the master process, and tasks are allocated to the worker processes, while the remaining tasks and computing resources are monitored. The tasks are independent for each protein pair and can be data parallelized.

In Azure cloud, we adopted the master–worker-type task dispatching in parallel, whereby one process was the master process and the remaining resources were used to execute multiple worker processes, and MPI communication was used to realize the task dispatching for the protein–protein interaction prediction. Unlike the case in a normal cluster-type computing environment, the distance between real machines in a cloud computing environment tends to be large, and MPI implementation is generally not considered as suitable. However, as MEGADOCK does not require heavy communication between tasks (worker processes), it was expected that the large-scale parallelization would not cause serious slowdowns.

Among the Azure instances available for HPC applications, we targeted A9, DS14, H16, and H16r as CPU instances with 16 CPU cores, as well as NC24 and NC24r as GPU instances equipped with 24 CPU cores and 4 GPU chips. The details of each instance are displayed in Table 1. For each process to be able to use one GPU, a task dispatching was performed to run four processes per instance (on a VM). That is, the number of CPU cores allocated to each task was 1/4 of the number of cores in each VM: four cores for CPU instances and six cores for GPU instances.

Table 1 Details of Azure instances used in study

Instance	CPU	# cores	Total DP peak (CPU)
DS14	Xeon E5-2660 @2.20 GHz × 2	16	281.6 GFlops
A9	Xeon E5-2670 @2.60 GHz × 2	16	332.8 GFlops
H16	Xeon E5-2667v3 @3.20 GHz × 2	16	691.2 GFlops
H16r	Xeon E5-2667v3 @3.20 GHz × 2	16	691.2 GFlops
NC24	Xeon E5-2690v3 @2.60 GHz × 2	24	883.2 GFlops
NC24r	Xeon E5-2690v3 @2.60 GHz × 2	24	883.2 GFlops

Instance	GPU	RAM	Network	Price (at March 2017)
DS14	N/A	112 GB	–	1.39 USD/h
A9	N/A	112 GB	RDMA supported	1.93 USD/h
H16	N/A	112 GB	–	1.75 USD/h
H16r	N/A	112 GB	RDMA supported	1.92 USD/h
NC24	Tesla K80 × 4 chips	1440 GB	–	4.32 USD/h
NC24r	Tesla K80 × 4 chips	1440 GB	RDMA supported	4.75 USD/h

Table 2 Results of MEGADOCK on Azure CPU instances (values in parentheses are the ratio of the calculation speed to H16)

Instance	50 instances	100 instances	Strong scaling
DS14	3283 s (0.47)	1696 s (0.48)	0.968
A9	2369 s (0.64)	1352 s (0.61)	0.876
H16	1527 s (1)	820 s (1)	0.931
H16r	1640 s (0.93)	953 s (0.86)	0.861

2.4 Experimental Settings

The dataset was the total of 59 protein heterodimeric complexes in the ZLAB protein–protein docking benchmark (version 1.0) [33]. The 59 heterodimers were divided, and all-to-all (cross) docking calculations were performed on the 59 receptor proteins and 59 ligand proteins.

3 Results and Discussion

3.1 MEGADOCK on Multiple CPU Instances

The results of the parallel execution of MEGADOCK on 50 and 100 instances using the CPU instances DS14, A9, H16, and H16r are presented in Table 2. The calculation time values were the median values measured three times. In this case, strong scaling was the value calculated as $\text{strong scaling} = (T_{50}/T_{100})/(100/50)$ when the computation times of 50 and 100 instances were T_{50} and T_{100} , respectively.

The experimental results demonstrated that the computation using the H16 instance was the fastest, followed by H16r, A9, and DS14. This ordering is

naturally corresponding to the order of CPU performance (total DP peak) presented in Table 1.

When 100 H16 instances (1600 CPU cores) were used, the calculation was completed in 820 s. This was the speed at which protein–protein docking calculations could be performed at 255 pairs per minute.

The calculation for the H16r instance was slightly slower than that for H16. The H16r is an instance that can use the RDMA network interface and exhibits higher communication performance than the H16, but MEGADOCK achieves higher performance even without RDMA network. An RDMA network may not be necessary for many bioinformatics applications in which data parallelization is possible. Moreover, as an instance with an RDMA network is more expensive than an instance without it, it is more reasonable not to use an RDMA network from a cost perspective.

The strong scaling was greater than 0.85 in the range of this measurement in all instances.

3.2 MEGADOCK on Multiple GPU Instances

Using the GPU instances NC24 and NC24r, we measured the computation times with using 5, 10, 20, and 40 instances. Figure 2 presents the measured calculation times and speed improvement rates. Owing to the limit of Microsoft Azure on the number of maximum concurrent GPUs (quota limit), the maximum number of allocated instances was 40. In the comparison between the NC24 and NC24r, the NC24r with an RDMA network slightly outperformed the NC24 in terms of speed, but the difference was very small. As with the CPU instance, the GPU instance would not require an RDMA network for this application.

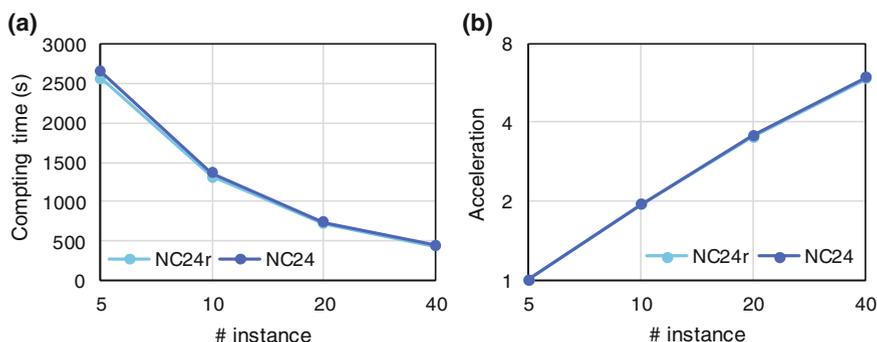


Fig. 2 Results of calculation time measurements on GPU instances: (a) computation time for each number of instances and (b) computation speed ratio with respect to five instances

NC24 is discussed below. When using 40 instances of NC24 (960 CPU cores and 160 GPUs), the calculation was completed within 448 s. This was faster than the result for the CPU instance indicated in Table 2 (H16: 1600 CPU cores) and enabled 466 pairs of protein–protein docking to be performed per minute. For strong scaling, the parallelization efficiency of 20 instances was 0.89 for five instances, which was similar to that of the CPU instances. However, when 40 instances were used, the speed improvement was only 5.91-fold faster than that of 5 instances, with a strong scaling value of 0.74.

3.3 Which Instance Should Be Used from a Cost Perspective?

3.3.1 CPU Instance

According to the comparison of CPU instances, the computation speed of the H16 instance was the most favorable. Comparing the H16 with the less expensive DS14, the speed improvement ratio was $1696 \text{ s}/820 \text{ s} = 2.07$. The price ratio between H16 (1.75 USD/h) and DS14 (1.39 USD/h) was $1.75 \text{ USD}/1.39 \text{ USD} = 1.26$. As a result, it is more reasonable to use the H16 than the DS14, as the value of the speed improvement ratio is larger than the price ratio.

Both A9 and H16r are slightly more expensive because they have an RDMA network, but MEGADOCK does not need to use these instances because no increase obtained in the computation speed when using an RDMA network. When using applications that require a powerful network, we recommend the H16r, which is approximately the same price as the A9 but provides higher CPU performance.

3.3.2 GPU Instance

A significant increase in the speed was achieved when using the GPU instance. However, unlike H16 and DS14, NC24 has 24 CPU cores, making a direct comparison difficult. In the following, we consider the maximum measurements at H16 (100 instances, 1600 cores, and 820 s) and NC24 (40 instances, 960 cores and 160 GPUs, and 448 s) in terms of the cost. Table 3 provides a summary of these results. In Table 3, the total fee was calculated by ignoring the time required for factors such as VM deployment and assuming that the product of {calculation time \times number of instances} used was the total cloud usage time.

Consequently, the same calculation could be performed for 21.5 USD for NC24, compared to 39.9 USD for H16. NC24 has a shorter execution time and is almost twice as advantageous in terms of usage fees. For GPU-enabled applications, the use of GPU instances offers the potential to yield computational results rapidly and inexpensively, and active consideration thereof is recommended.

Table 3 Summary of results for H16 and NC24 instances

Instance	# inst.	CPU cores	GPUs	Time	Price (1 inst.)	Total fee ^a
H16	100	1600	N/A	820 s	1.75 USD/h	39.9 USD
NC24	40	960	160	448 s	4.32 USD/h	21.5 USD

^a The total fee was obtained by Price \times Time (h) \times # inst

4 Conclusions

We constructed a computing environment for large-scale protein–protein docking calculations with the MEGADOCK software on the public cloud of Microsoft Azure and performed large-scale parallel calculations on approximately 1000 GPUs. We found that MEGADOCK provided the fastest GPU computation on the NC24 instance and the cloud computing cost was lower than that of using CPU instances.

Large-scale data analysis with MEGADOCK requires high CPU and GPU performance, but does not require high communication performance. For bioinformatics applications similar in properties to MEGADOCK, it would be most cost-effective to use the NC24 instance or the similar instance without high-bandwidth network, like RDMA, as in this study.

The use of the public cloud environment is advantageous owing to the portability and reproducibility of computing applications, and it allows for the rapid construction of large-scale applications such as the one investigated in this study. The cloud environment is particularly useful in applications such as pipeline software, in which various tools are intricately interrelated. We have already developed a system to enable MEGADOCK computation by constructing a container virtualization environment on the cloud [34]. In addition to the protein–protein docking calculations demonstrated in this study, various other bioinformatics applications operating on the public cloud will certainly contribute to accelerating the research in this field.

Acknowledgments The authors thank Mr. Hiroyuki Sato of IMSBIO, Co., Ltd. for his technical support in the development on Microsoft Azure. This work was partially supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI (18K18149, 20H04280), Core Research for Evolutional Science and Technology (CREST) “Extreme Big Data” (Grant No. JPMJCR1303) from the Japan Science and Technology Agency (JST), the Platform Project for Supporting Drug Discovery and Life Science Research (Basis for Supporting Innovative Drug Discovery and Life Science Research (BINDS)) (Grant No. JP18am0101112) from the Japan Agency for Medical Research and Development (AMED), Microsoft Business Investment Funding from Microsoft Corporation, and Leave a Nest Grant from Leave a Nest Co., Ltd. This work was partially conducted as research activities of AIST-Tokyo Tech Real World Big-Data Computation Open Innovation Laboratory (RWBC-OIL). The authors thank Editage (www.editage.com) for English language editing.

References

1. I.A.T. Hashem, I. Yaqoob, N.B. Anuar et al., The rise of “big data” on cloud computing: Review and open research issues. *Inf. Syst.* **47**, 98–115 (2015). <https://doi.org/10.1016/j.is.2014.07.006>
2. R. Tudoran, A. Costan, G. Antoniu et al., A performance evaluation of Azure and Nimbus clouds for scientific applications, in *Proc CloudCP'12* (2012), pp. 1–6. <https://doi.org/10.1145/2168697.2168701>
3. A. O'Driscoll, J. Daugeleite, R.D. Sleator, ‘Big data’, Hadoop and cloud computing in genomics. *J. Biomed. Inform.* **46**(5), 774–781 (2013). <https://doi.org/10.1016/j.jbi.2013.07.001>
4. V. Sobeslav, P. Maresova, O. Krejcar et al., Use of cloud computing in biomedicine. *J. Biomol. Struct. Dyn.* **34**(12), 2688–2697 (2016). <https://doi.org/10.1080/07391102.2015.1127182>
5. J. Karlsson, O. Torreno, D. Ramet et al., Enabling large-scale bioinformatics data analysis with cloud computing, in *Proc IEEE ISPA2012* (2012), pp. 640–645. <https://doi.org/10.1109/ISPA.2012.95>
6. H.P. Shanahan, A.M. Owen, A.P. Harrison, Bioinformatics on the cloud computing platform azure. *PLoS One* **9**(7), e102642 (2014). <https://doi.org/10.1371/journal.pone.0102642>
7. Ekanayake J, Gunarathne T, Qiu J (2011) Cloud Technologies for Bioinformatics Applications. *IEEE Trans Parallel Distrib Syst* **22**(6), 998–1011. <https://doi.org/10.1109/TPDS.2010.178>
8. A. Matsunaga, M. Tsugawa, J. Fortes, CloudBLAST: Combining MapReduce and virtualization on distributed resources for bioinformatics applications, in *Proc IEEE eScience2008* (2008), pp. 222–229. <https://doi.org/10.1109/eScience.2008.62>
9. W. Lu, J. Jackson, R. Barga, AzureBlast: A case study of developing science applications on the cloud, in *Proc ACM HPDC'10* (2010), pp. 413–420. <https://doi.org/10.1145/1851476.1851537>
10. T. Gunarathne, T.-L. Wu, J.Y. Choi et al., Cloud computing paradigms for pleasingly parallel biomedical applications. *Concurr. Comput. Pract. Exp.* **23**(17), 2338–2354 (2011). <https://doi.org/10.1002/cpe.1780>
11. D. Mrozek, B. Małysiak-Mrozek, A. Kłapciński, Cloud4Psi: cloud computing for 3D protein structure similarity searching. *Bioinformatics* **30**(19), 2822–2825 (2014). <https://doi.org/10.1093/bioinformatics/btu389>
12. D. Mrozek, T. Kutyla, B. Małysiak-Mrozek, Accelerating 3D protein structure similarity searching on microsoft azure cloud with local replicas of macromolecular data, in *Proc PPAM2015, LNCS 9574* (2016), pp. 254–265. https://doi.org/10.1007/978-3-319-32152-3_24
13. D. Mrozek, P. Gosk, B. Małysiak-Mrozek, Scaling *Ab Initio* Predictions of 3D protein structures in microsoft azure cloud. *J Grid Comput* **13**, 561–585 (2015). <https://doi.org/10.1007/s10723-015-9353-8>
14. B.T. Moghadam, J. Alvarsson, M. Holm et al., Scaling predictive modeling in drug development with cloud computing. *J. Chem. Inf. Model* **55**(1), 19–25 (2015). <https://doi.org/10.1021/ci500580y>
15. Z. Farkas, P. Kacsuk, T. Kiss et al., AutoDock gateway for molecular docking simulations in cloud systems, in *Cloud Computing with e-Science Applications* (2015), pp. 217–236. <https://doi.org/10.1201/b18021-11>
16. R. De Paris, D.A.D. Ruiz, O.N. de Souza, A cloud-based workflow approach for optimizing molecular docking simulations of fully-flexible receptor models and multiple ligands, in *Proc IEEE CloudCom2015* (2015), pp. 495–498. <https://doi.org/10.1109/CloudCom.2015.43>
17. P. Hodor, A. Chawla, A. Clark et al., cl-dash: rapid configuration and deployment of Hadoop clusters for bioinformatics research in the cloud. *Bioinformatics* **32**(2), 301–303 (2015). <https://doi.org/10.1093/bioinformatics/btv553>
18. J. Qiu, J. Ekanayake, T. Gunarathne et al., Hybrid cloud and cluster computing paradigms for life science applications. *BMC Bioinform.* **11**, S3 (2010). <https://doi.org/10.1186/1471-2105-11-S12-S3>
19. H.A. Hassan, S.A. Mohamed, W.M. Sheta, Scalability and communication performance of HPC on Azure Cloud. *Egypt Inform. J.* **17**(2), 175–182 (2016). <https://doi.org/10.1016/j.eij.2015.11.001>

20. Y. Matsuzaki, N. Uchikoga, M. Ohue et al., Rigid-docking approaches to explore protein-protein interaction space. *Adv. Biochem. Eng. Biotechnol.* **160**, 33–55 (2017). https://doi.org/10.1007/10_2016_41
21. E. Chermak, A. Petta, L. Serra et al., CONSRANK: a server for the analysis, comparison and ranking of docking models based on inter-residue contacts. *Bioinformatics* **31**(9), 1481–1483 (2015). <https://doi.org/10.1093/bioinformatics/btu837>
22. G. Launay, M. Ohue, J.P. Santero et al., Rescoring ensembles of protein-protein docking poses using consensus approaches (2020). *bioRxiv* 2020.04.24.059469. <https://doi.org/10.1101/2020.04.24.059469>
23. A. Lopes, S. Sacquin-Mora, V. Dimitrova et al., Protein-protein interactions in a crowded environment: an analysis via cross-docking simulations and evolutionary information. *PLoS Comput. Biol.* **9**, e1003369 (2013). <https://doi.org/10.1371/journal.pcbi.1003369>
24. M. Ohue, T. Shimoda, S. Suzuki et al., MEGADOCK 4.0: an ultra-high-performance protein-protein docking software for heterogeneous supercomputers. *Bioinformatics* **30**(22), 3281–3283 (2014). <https://doi.org/10.1093/bioinformatics/btu532>
25. T. Hayashi, Y. Matsuzaki, K. Yanagisawa et al., MEGADOCK-Web: an integrated database of high-throughput structure-based protein-protein interaction predictions. *BMC Bioinform.* **19**, 62 (2018). <https://doi.org/10.1186/s12859-018-2073-x>
26. M.N. Wass, G. Fuentes, C. Pons et al., Towards the prediction of protein interaction partners using physical docking. *Mol. Syst. Biol.* **7**, 469 (2011). <https://doi.org/10.1038/msb.2011.3>
27. C. Zhang, B. Tang, Q. Wang et al., Discovery of binding proteins for a protein target using protein-protein docking-based virtual screening. *Proteins* **82**(10), 2472–2482 (2014). <https://doi.org/10.1002/prot.24611>
28. E. Karaca, A.M.J.J. Bonvin, A multidomain flexible docking approach to deal with large conformational changes in the modeling of biomolecular complexes. *Structure* **19**(4), 555–565 (2011). <https://doi.org/10.1016/j.str.2011.01.014>
29. Y. Matsuzaki, N. Uchikoga, M. Ohue et al., MEGADOCK 3.0: a high-performance protein-protein interaction prediction software using hybrid parallel computing for petascale supercomputing environments. *Source Code Biol. Med.* **8**, 18 (2013). <https://doi.org/10.1186/1751-0473-8-18>
30. M. Ohue, Y. Matsuzaki, N. Uchikoga et al., MEGADOCK: An all-to-all protein-protein interaction prediction system using tertiary structure data. *Protein Pept. Lett.* **21**(8), 766–778 (2014). <https://doi.org/10.2174/09298665113209990050>
31. E. Katchalski-Katzir, I. Shariv, M. Eisenstein et al., Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. *Proc. Natl. Acad. Sci. U. S. A.* **89**(6), 2195–2199 (1992). <https://doi.org/10.1073/pnas.89.6.2195>
32. T. Shimoda, S. Suzuki, M. Ohue et al., Protein-protein docking on hardware accelerators: comparison of GPU and MIC architectures. *BMC Syst. Biol.* **9**(Suppl 1), S6 (2015). <https://doi.org/10.1186/1752-0509-9-S1-S6>
33. R. Chen, J. Mintseris, J. Janin et al., A protein-protein docking benchmark. *Proteins* **52**, 88–91 (2003). <https://doi.org/10.1002/prot.10390>
34. K. Aoyama, Y. Yamamoto, M. Ohue et al., Performance evaluation of MEGADOCK protein-protein interaction prediction system implemented with distributed containers on a cloud computing environment, in *Proc PDPTA'19* (2019), pp. 175–181

HoloMol: Protein and Ligand Visualization System for Drug Discovery with Augmented Reality



Atsushi Koyama, Shingo Kawata, Wataru Sakamoto, Nobuaki Yasuo, and Masakazu Sekijima

1 Introduction

The creation of new drugs is generally considered to be time-consuming and expensive. Research in drug discovery involves the identification of disease-causing target proteins, followed by the search for ligands, which are compounds that specifically bind to the target protein and inhibit its action. Then, preclinical studies, such as ligand structure optimization, confirmation of cytotoxicity, and animal experiments, are performed before proceeding to the clinical trials. According to the US data, it would take 12–14 years and approximately \$2.6 billion to introduce a single drug in the market [1]. Various methods and approaches have been developed to reduce the time and cost of acquiring new compounds and to search for promising drug candidates [2, 3]. Therefore, new technologies to improve the efficiency of drug discovery are being explored on a global scale, and researchers have shown considerable interest in efficient drug discovery methods using information technology to reduce the costs involved.

In the post-genome era, CADD is considered to be one of the most efficient methods to achieve the abovementioned goals. It uses large-scale genome sequence information, protein structure information, and small molecule compound information to identify the target protein, find the ligand, absorb, distribute, metabolize,

A. Koyama · S. Kawata · W. Sakamoto · M. Sekijima (✉)
School of Computing, Tokyo Institute of Technology, Yokohama, Japan
e-mail: koyama@cbi.c.titech.ac.jp; kawata@cbi.c.titech.ac.jp; sakamoto@cbi.c.titech.ac.jp;
sekijima@c.titech.ac.jp

N. Yasuo
Academy for Convergence of Materials and Informatics, Tokyo Institute of Technology, Tokyo, Japan
e-mail: yasuo@c.titech.ac.jp

and so on [4–14]. Additionally, the use of CADD is expected to reduce the cost of drug discovery by up to 50% [15]. The three-dimensional (3D) structures of proteins, which are targets for the development of therapeutic drugs, are elucidated and accumulated as information every year. In fact, the number of 3D structures registered in the Protein Data Bank (PDB) [16], a 3D protein structure database, has increased by approximately three times in the last decade. The main methods of drug design are structure-based drug design (SBDD) [17, 18], in which compounds are explored based on protein structure information, and ligand-based drug design (LBDD) [19], in which compounds are explored using properties, such as the activity of a compound (ligand) that binds to a specific protein and the concept (pharmacophore) of functional groups, which enable the interaction between a drug target and their relative steric configurations.

To date, virtual reality (VR) and augmented reality (AR) head-mounted display (HMD)-type devices have been developed. As a result, VR and AR are expected to become more familiar in our daily lives in the near future. In recent years, VR and AR have already been introduced in the medical field as a simulation system and support system for surgery. However, in the case of proteins, it is essential for drug discovery, particularly SBDD, to deepen our understanding of the stereoscopic structural information, such as protein pockets, by drawing 3D images using VR and AR.

A system to display proteins and compounds in a VR space has been developed using VR devices [20–22]. For example, Molecular Rift can manipulate proteins by detecting the position, shape, and movement of the user's hand using Oculus Rift and Kinect, a type of VR device. However, with Molecular Rift, it is difficult to observe the displayed protein while interacting with the surroundings because the view is completely blocked by the head-mounted display; it is also difficult to observe the protein while referring to the material at hand in a meeting. Moreover, it is difficult to carry the system around because it needs to be connected to a PC or Kinect. In this study, we develop a 3D rendering system for proteins and compounds using a head-mounted display, which can operate by itself without a PC. Our goal is to develop not only a protein 3D display device but also an integrated system that is more useful for drug discovery by linking to the cloud.

2 Development Environment

2.1 HoloLens

In this study, Microsoft HoloLens (Fig. 1) was selected as the visualization device. Microsoft HoloLens is an HMD-type wearable computer developed by Microsoft. The sensor cameras on both sides scan the user's vision, map their environment in real time, and project holograms onto a transparent display based on the scanned data; thus, HoloLens achieves augmented reality [23, 24].

Fig. 1 Microsoft HoloLens**Table 1** Basic specification of HoloLens

Items	Details
Speakers	Built-in Speakers
Wireless LAN	Wi-Fi 802.11ac
Bluetooth	Bluetooth 4.0 LE
OS	Windows 10
CPU	Intel 32-bit architecture (1 GHz)
GPU	Custom-built Microsoft Holographic Processing Unit
Storage	64 GB flash
RAM	2 GB
Battery	2–3 h
Power supply	2.5 A/5.2 V
Weight	579 g

In comparison with other VR and AR devices, HoloLens is a wearable computer with an OS and CPU integrated in the device itself and, hence, can operate as a standalone device. In conventional head-mounted display-type VR devices, the video processing is performed by the PC, and display is enabled by the device. Therefore, when moving around the floor with a VR head-mounted display, the user needs to use a backpack-type PC or may require assistance from others. In contrast, HoloLens can operate independently. Additionally, HoloLens can share the same hologram with other HoloLens. Unlike conventional VR devices, the open field of view makes it easy to communicate with others and view the materials at hand simultaneously. Furthermore, HoloLens is easy to use in meetings, and the 3D molecule can be used to communicate with others and refer to materials. In this study, the application on HoloLens was developed using Unity, which is widely used in VR and AR devices. However, because the hologram is displayed on a transparent display, the visibility of the hologram is affected by the brightness of the reality. Moreover, the viewing angle of HoloLens is approximately 35° , which is narrower than that of other VR devices [25]. The detailed specification of the HoloLens is presented in Table 1.

2.2 Universal Windows Platform

Universal Windows Platform (UWP), introduced in Windows 10, is a computing platform that integrates applications running on a variety of devices into a single framework. By setting the target device, users can develop UWP applications with device-specific functions by adding a device-specific API set to the basic API set that is common to all devices. The UWP app is the only application that is compatible with HoloLens, whereas traditional desktop PC apps, including Windows Forms apps, are incompatible. Although this UWP application has enhanced security, because of the restrictions on the use of local resources, it requires the end user's permission to access files on the client PC and connected devices and cannot call other running processes or external processes, such as traditional desktop applications. Therefore, in this study, we were required to overcome these problems, for example, by reading the protein data.

2.3 Unity

Unity is a game engine developed by Unity Technologies, which can be used on a wide range of platforms, such as Mac OS, iOS, Android, and UWP. In the development of holographic applications, visual information is a key aspect, and unlike XAML, Unity allows development while checking the execution of the application. Therefore, Unity is suitable for creating applications for HoloLens.

The development environment is summarized in Table 2.

3 HoloMol

3.1 Overview

A schematic diagram of the HoloMol system is illustrated in Fig. 2. The proteins to be observed are downloaded from the PDB in pdb format and converted into 3D models in Collada format using PyMol. By using the Collada format, data

Table 2 Development environment

Environment	Version
OS	Windows 10 Pro
Game engine	Unity 2017.2.1f1
Development language	C # 4.0
Execution environment	.NET Framework 3.5
Integrated development environment	Visual Studio 2017
HMD	Microsoft HoloLens

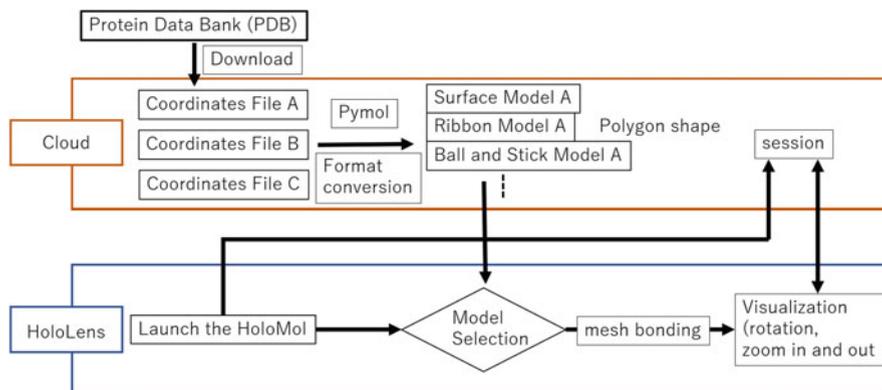


Fig. 2 System overview

can be exchanged between various digital content creation tools in the cloud and can be handled by Unity. The converted Collada files are stored in the cloud and downloaded to HoloLens using the user interface of the UWP app as a trigger for the actual protein observation. To reduce the load of drawing on HoloLens, the meshes are combined as described below. Currently, the process of downloading, converting, and uploading the PDB files is performed manually; however, we are implementing a function to perform this on the cloud.

3.2 Visualization Details

Holographic applications have various restrictions, such as being unable to call an external process as described above, because of UWP. There are three main ways to call external objects that are not placed in the initial state when the application is launched:

1. Stream additional assets using the AssetBundle and WWW class, and instantiate them at runtime.
2. Set a path by Resources.Load and load the assets in the Resources folder.
3. Connect to an external URL using the WWW class, directly load objects of a specific format, and place them in the scene.

Each method has its own advantages and disadvantages, as presented in Table 3.

Method 1 is widely used in smartphone applications as it can reduce the data capacity to be transmitted and received. Method 2 enables the suppression of memory usage. Method 3 only needs to connect to an external URL; hence no effort is required in preparing to use the application. However, each method has its disadvantages, and methods 1 and 2 are found to be unsuitable for this system.

Table 3 Comparison of methods of calling external resource objects

	Advantages	Disadvantages
1	It is possible to suppress the data capacity to be transmitted and received	Unity environment is required when a new model is added to create the AssetBundle
2	It is possible to reduce memory usage	Objects need to be placed in the Resources folder
3	It takes less time and effort in adding new models in comparison with other methods	Model reading speed is slow and the format of the object is limited

In method 1, the number of structural data currently registered in the PDB is approximately 140,000, which is a large number; therefore, it is not realistic to prepare and manage all the structural data as an AssetBundle. Additionally, the PC-installed Unity is required to create the AssetBundle, and it is not desirable to have users create the AssetBundle because it leads to deterioration of usability. In method 2, all the possible 3D models must be stored in the project's Resource folder; therefore, it is not realistic to have approximately 140,000 types of structural data in the folder at a time. HoloMol uses method 3, which does not incorporate the 3D model into HoloLens as an asset beforehand, but connects it to an external URL during playback and places the object directly in the scene.

3.3 Object Integration and Mesh Binding

Unity requires the material, mesh, and shader information to draw an object. The material information comprises the base image of the object surface, surface color, and other information. A mesh is made up of a set of polygonal surfaces called polygons. The mesh information comprises the positional and normal information of the polygons. The shader draws the object according to the mesh information. The computational cost of executing the drawing instructions issued by the shader is high, causing a heavy operation when handling a model with a large number of objects. A 3D model, such as the protein and compound complex used in this study, may consist of tens of thousands of objects, and because all objects contain material and mesh information, we can expect to reduce the weight of processing by integrating the objects. Because the protein model comprises many objects, it is estimated that some 3D models may be extremely slow. In this study, the mesh information of the drawn protein model is integrated at runtime to reduce the number of objects that make up the model and to reduce the weight of the operation. Mesh binding is executed using a C# script. In Unity, there is a limit to the number of vertices that can be handled by a single object; therefore, we combine meshes so that they do not exceed the limit.

3.4 Evaluation of Mesh Binding

The protein models used in this study are 1FKB, 1XL2, 3RZE, and 5GGR. In the experiment, the 3D protein 3D model was placed at a distance of 1.5 m from the device, adjusted to a scale that allows the entire model to be seen, and then the frame rate was measured when a rotation operation was applied. The frame rate values obtained by the Mixed Reality Toolkit Visual Profiler of MRTK v2 are used for the measurement.

4 Results

In HoloMol, the 3D structures of proteins and compounds can be freely rotated and expanded. Figure 3 depicts a hologram that is actually seen through HoloLens. Figure 4 illustrates a graph comparing the number of frames drawn per second (FPS) for each protein and model. Even for the same protein, particularly in the stick model, the computational load is high because the objects are divided into smaller parts. Therefore, we can see that the surface/cartoon model runs faster than the stick model without object combination.

Figure 5 depicts the models with and without mesh binding. We can confirm that the original model is not compromised by mesh binding. The changes in the frame rate of the stick model of a protein drawn on HoloLens with or without mesh binding are depicted in Fig. 6. The frame rate without mesh binding was below 30 FPS for all models.

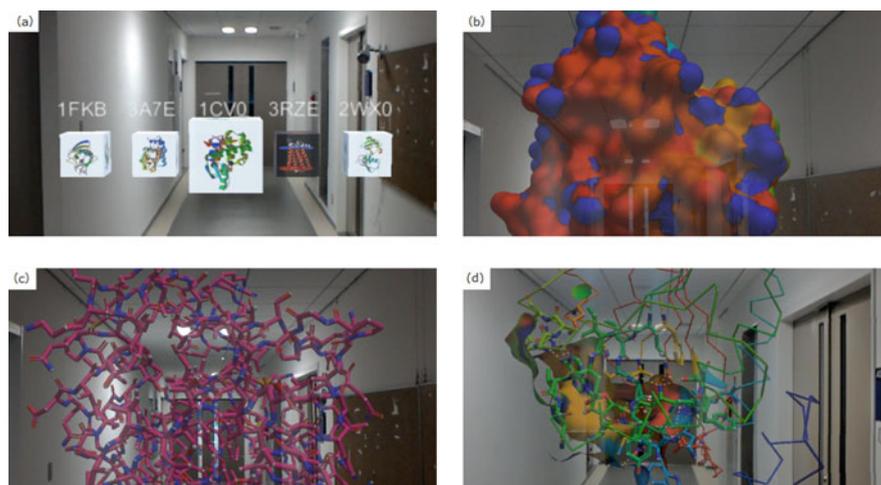


Fig. 3 Images seen through HoloLens: (a) UI for switching models. (b) Surface model. (c) Stick model. (d) Ligand binding site

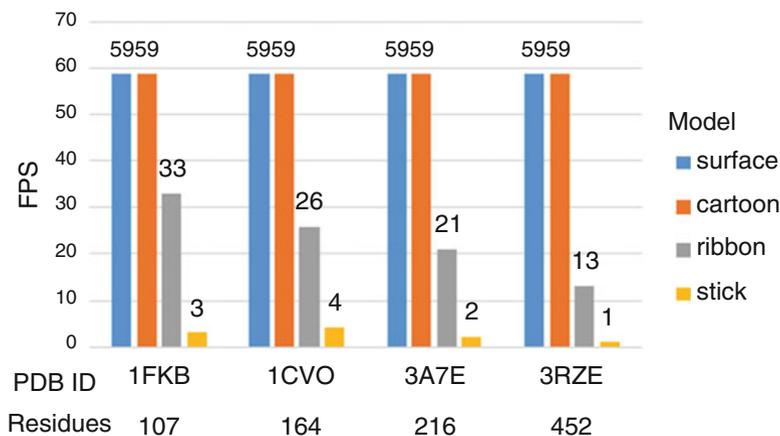


Fig. 4 Comparison of FPS in each model and protein

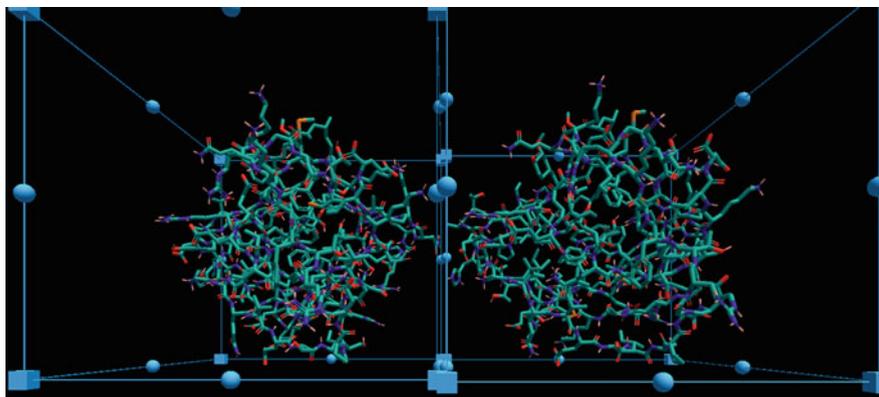


Fig. 5 Mesh-bonded model (left) and original model (right)

Table 4 presents the number of objects after joining of the stick model for each protein; in HoloMol, the number of vertices in a mesh is set to 65,535, and the number of objects is reduced. In general, a frame rate of at least 30 FPS is required for comfortable operation. HoloLens set the frame rate limit to 60 FPS, and 1FKB, 1XL2, and 3RZE were set at the upper limit. 5GGR recorded a frame rate of 55 FPS. Although the protein model with a large number of residues demonstrated a decrease in FPS, the present method seems to have demonstrated a certain effect. In the absence of mesh binding, low values were recorded for all the stick models of proteins. We demonstrated that mesh binding improved the frame rate for all the proteins tested in this experiment. However, even when mesh binding was performed, we observed that the frame rate was below 30 FPS when the number of residues exceeded 1900. Although the frame rate decreased as the number of

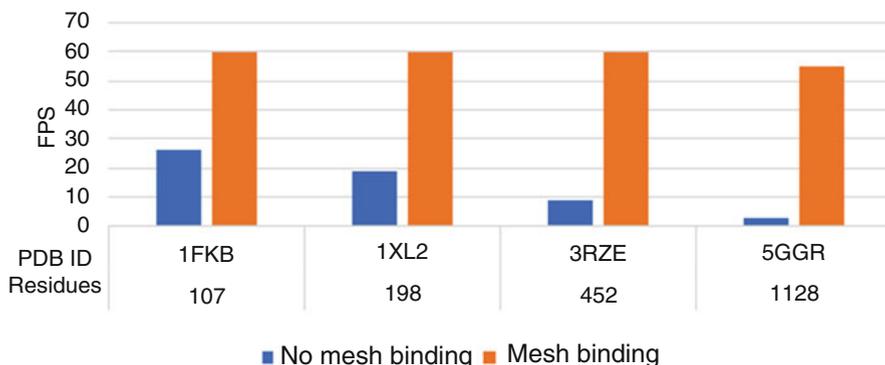


Fig. 6 Effect of mesh binding

Table 4 Number of objects after mesh binding

PDB ID	Before	After
1FKB	2076	1
1XL2	2962	1
3RZE	6665	2
5GGR	15,756	5

residues increased, in this study, it was confirmed that the protein model could be smoothly drawn up to a protein model with a residue number of approximately 1000. The number of residues increased; in this study, it was confirmed that the protein model could be smoothly drawn up to a protein model with a residue number around 1000.

5 Conclusions

In this study, we developed HoloMol, a drug discovery support system that visualizes the 3D structures of proteins and compounds, using Microsoft HoloLens, which can run on its own and is linked to the cloud. HoloMol enables us to observe holograms of proteins and compounds while referring to the materials at hand, which was not possible with VR-based systems. Furthermore, we aimed to make it possible for multiple people to share the hologram, walk around and peer into the surroundings, understand the 3D structure of the protein and the binding mode of the protein and ligand, and discuss ligand optimization.

Microsoft HoloLens has a viewing angle of approximately 35° , which implies that the user cannot see through some 3D models on the display without adjusting the model size. However, its successor, the Microsoft HoloLens2, has a wider viewing angle of approximately 52° with new features, such as eye-tracking and five-finger recognition, which allow for more intuitive operation. Additionally, the

ease of wearing the device for long durations, which was one of the issues in HoloLens, is also achieved by changing the hardware. In future research, we intend to develop a system that is more comfortable for users by porting this system to HoloLens2 and adding functions, such as manipulation of protein models using fingers on both hands.

Acknowledgments This work was partially supported by the Research Complex Program “Well-being Research Campus: Creating New Values Through Technological and Social Innovation” from the Japan Science and Technology Agency (JST), the Platform Project for Supporting Drug Discovery and Life Science Research (Basis for Supporting Innovative Drug Discovery and Life Science Research (BINDS)) from AMED under Grant Number JP20am0101112, and the Japanese Society for the Promotion of Science (JSPS) KAKENHI under Grant Number 20H00620.

References

1. A. Mullard, New drugs cost US \$2.6 billion to develop. *Nat. Rev. Drug Discov.* **13**(12), 877 (2014)
2. S.J. Haggarty, T.U. Mayer, D.T. Miyamoto, R. Fathi, R.W. King, T.J. Mitchison, S.L. Schreiber, Dissecting cellular processes using small molecules: identification of colchicine-like, taxol-like and other small molecules that perturb mitosis. *Chem. Biol.* **7**, 1074–5521 (2000)
3. K. Young, S. Lin, L. Sun, E. Lee, M. Modi, S. Hellings, M. Husbands, B. Ozenberger, R. Franco, Identification of a calcium channel modulator using a high throughput yeast two-hybrid screen. *Nat. Biotechnol.* **16**, 1546–1696 (1998)
4. R. Yoshino, N. Yasuo, D.K. Inaoka, Y. Hagiwara, K. Ohno, M. Orita, M. Inoue, T. Shiba, S. Harada, T. Honma et al., Pharmacophore modeling for anti-chagas drug design using the fragment molecular orbital method. *PLoS One* **10**(5), e0125829 (2015)
5. S. Chiba, K. Ikeda, T. Ishida, M.M. Gromiha, Y. Taguchi, M. Iwadata, H. Umeyama, K.Y. Hsin, H. Kitano, K. Yamamoto, N. Sugaya, K. Kato, T. Okuno, G. Chikenji, M. Mochizuki, N. Yasuo, R. Yoshino, K. Yanagisawa, T. Ban, R. Teramoto, C. Ramakrishnan, A.M. Thangakani, D. Velmurugan, P. Prathipati, J. Ito, Y. Tsuchiya, K. Mizuguchi, T. Honma, M. Sekijima, Identification of potential inhibitors based on compound proposal contest: Tyrosine-protein kinase yes as a target. *Sci. Rep.* **5**, 1–13 (2015)
6. S. Chiba, T. Ishida, K. Ikeda, M. Mochizuki, R. Teramoto, Y. Taguchi, M. Iwadata, H. Umeyama, C. Ramakrishnan, A.M. Ramakrishnan, D. Velmurugan, M.M. Gromiha, T. Okuno, K. Kato, S. Minami, G. Chikenji, S.D. Suzuki, K. Yanagisawa, W.H. Shin, D. Kihara, K.Z. Yamamoto, Y. Moriwaki, N. Yasuo, R. Yoshino, S. Zozulya, P. Borysko, R. Stavniichuk, T. Honma, T. Hirokawa, Y. Akiyama, M. Sekijima, An iterative compound screening contest method for identifying target protein inhibitors using the tyrosine-protein kinase yes. *Sci. Rep.* **7**(1), 12038 (2017)
7. C. Ramakrishnan, A.M. Thangakani, D. Velmurugan, D.A. Krishnan, M. Sekijima, Y. Akiyama, M.M. Gromiha, Identification of type I and type II inhibitors of c-yes kinase using in silico and experimental techniques. *J. Biomol. Struct. Dynam.* **36**(6), 1566–1576 (2017). <https://doi.org/10.1080/07391102.2017.1329098>
8. N. Arai, S. Yoshikawa, N. Yasuo, R. Yoshino, M. Sekijima, Compound property enhancement by virtual compound synthesis. *J. Bioinf. Comput. Biol.* **16**(03), 1840016 (2018). <https://doi.org/10.1142/s0219720018400164>
9. N. Yasuo, K. Watanabe, H. Hara, K. Rikimaru, M. Sekijima, Predicting strategies for lead optimization via learning to rank. *IPJSJ Trans. Bioinf.* **11**(0), 41–47 (2018). <https://doi.org/10.2197/ipsjtbio.11.41>

10. N. Wakui, R. Yoshino, N. Yasuo, M. Ohue, M. Sekijima, Exploring the selectivity of inhibitor complexes with bcl-2 and bcl-xl: A molecular dynamics simulation approach. *J. Mol. Graph. Model.* **79**, 166–174 (2018)
11. N. Yasuo, Y. Nakashima, M. Sekijima, CoDe-DTI: Collaborative deep learning-based drug-target interaction prediction, in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (IEEE, Piscataway, 2018), pp. 792–797
12. S. Chiba, M. Ohue, A. Gryniukova, P. Borysko, S. Zozulya, N. Yasuo, R. Yoshino, K. Ikeda, W.H. Shin, D. Kihara, M. Iwadate, H. Umeyama, T. Ichikawa, R. Teramoto, K.Y. Hsin, V. Gupta, H. Kitano, M. Sakamoto, A. Higuchi, N. Miura, K. Yura, M. Mochizuki, C. Ramakrishnan, A.M. Thangakani, D. Velmurugan, M.M. Gromiha, I. Nakane, N. Uchida, H. Hakariya, M. Tan, H.K. Nakamura, S.D. Suzuki, T. Ito, M. Kawatani, K. Kudoh, S. Takashina, K.Z. Yamamoto, Y. Moriwaki, K. Oda, D. Kobayashi, T. Okuno, S. Minami, G. Chikenji, P. Prathipati, C. Nagao, A. Mohsen, M. Ito, K. Mizuguchi, T. Honma, T. Ishida, T. Hirokawa, Y. Akiyama, M. Sekijima, A prospective compound screening contest identified broader inhibitors for sirtuin 1. *Sci. Rep.* **9**(1) (2019). <https://doi.org/10.1038/s41598-019-55069-y>
13. R. Yoshino, N. Yasuo, M. Sekijima, Molecular dynamics simulation reveals the mechanism by which the influenza cap-dependent endonuclease acquires resistance against baloxavir marboxil. *Sci. Rep.* **9**(1) (2019). <https://doi.org/10.1038/s41598-019-53945-1>
14. N. Yasuo, M. Sekijima, Improved method of structure-based virtual screening via interaction-energy-based learning. *J. Chem. Inf. Model.* **59**(3), 1050–1061 (2019). <https://doi.org/10.1021/acs.jcim.8b00673>
15. J.J. Tan, X.J. Cong, L.M. Hu, C.X. Wang, L. Jia, X.J. Liang, Therapeutic strategies underpinning the development of novel techniques for the treatment of HIV infection. *Drug Discov. Today* **15**(5), 186–197 (2010). <http://www.sciencedirect.com/science/article/pii/S1359644610000115>
16. P.W. Rose, B. Beran, C. Bi, W.F. Bluhm, D. Dimitropoulos, D.S. Goodsell, A. Prlic, M. Quesada, G.B. Quinn, J.D. Westbrook, J. Young, B. Yukich, C. Zardecki, H.M. Berman, P.E. Bourne, The RCSB protein data bank: redesigned web site and web services. *Nucleic Acids Res.* **39**(Database), D392–D401 (2010). <https://doi.org/10.1093/nar/gkq1021>
17. E. Lionta, G. Spyrou, D.K. Vassiliatis, Z. Cournia, Structure-based virtual screening for drug discovery: Principles, applications and recent advances. *Curr. Top. Med. Chem.* **14**(16), 1923 (2014)
18. S. Ghosh, A. Nie, J. An, Z. Huang, Structure-based virtual screening of chemical libraries for drug discovery. *Curr. Opin. Chem. Biol.* **10**(3), 194–202 (2006)
19. C. Acharya, A. Coop, J.E. Polli, A.D. MacKerell Jr, Recent advances in ligand-based drug design: relevance and utility of the conformationally sampled pharmacophore approach. *Curr. Comput. Aided Drug Des.* **7**(1), 10 (2011)
20. A. Anderson, Z. Weng, VRDD: applying virtual reality visualization to protein docking and design. *J. Mol. Graph. Model.* **17**(3–4), 180–186 (1999). [https://doi.org/10.1016/s1093-3263\(99\)00029-7](https://doi.org/10.1016/s1093-3263(99)00029-7)
21. M. Norrby, C. Grebner, J. Eriksson, J. Boström, Molecular rift: Virtual reality for drug designers. *J. Chem. Inf. Model.* **55**(11), 2475–2484 (2015). <https://doi.org/10.1021/acs.jcim.5b00544>
22. C.M. Nakano, E. Moen, H.S. Byun, H. Ma, B. Newman, A. McDowell, T. Wei, M.Y. El-Naggar, iBET: Immersive visualization of biological electron-transfer dynamics. *J. Mol. Graph. Model.* **65**, 94–99 (2016). <https://doi.org/10.1016/j.jmkgm.2016.02.009>
23. R.T. Azuma, A survey of augmented reality. *Presence Teleop. Virtual Environ.* **6**(4), 355–385 (1997). <https://doi.org/10.1162/pres.1997.6.4.355>
24. S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P.L. Davidson, S. Khamis, M. Dou, V. Tankovich, C. Loop, Q. Cai, P.A. Chou, S. Mennicken, J. Valentin, V. Pradeep, S. Wang, S.B. Kang, P. Kohli, Y. Lutchnyn, C. Keskin, S. Izadi, Holoportation, in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (ACM, New York, 2016). <https://doi.org/10.1145/2984511.2984517>
25. P. Saarikko, P. Kostamo, Waveguide, patent No. US20160231568A1, Filed Feb. 9th, 2015, Issued Aug. 11th, 2016

Leave-One-Element-Out Cross-Validation for Band Gap Prediction of Halide Double Perovskites



Hiroki Igarashi, Nobuaki Yasuo, and Masakazu Sekijima

1 Introduction

Conventional materials development requires repeated experiments and assessments based on the knowledge and experience of researchers. It often takes a long period of time, such as 20 years or more, before it can be put to practical use [1]. In order to reduce the cost of materials development, the field of materials informatics, in which materials development is combined with information technology and computational science methods, has been popular in recent years. In materials informatics, it is possible to search for materials efficiently by simulating or predicting material candidates with necessary properties and deciding which materials to synthesize, instead of synthesizing them to investigate their properties, independent from individualization.

Machine learning-based molecular property prediction and computational compound enumeration have been widely used in cheminformatics, especially in computational drug discovery [2–7]. These methods have been transferred to materials informatics. Notable difference between computational drug discovery

H. Igarashi

Department of Computer Science, Tokyo Institute of Technology, Yokohama, Kanagawa, Japan

N. Yasuo

Academy for Convergence of Materials and Informatics, Tokyo Institute of Technology, Meguro, Tokyo, Japan

e-mail: yasuo@cbi.c.titech.ac.jp

M. Sekijima (✉)

Department of Computer Science, Tokyo Institute of Technology, Yokohama, Kanagawa, Japan

Academy for Convergence of Materials and Informatics, Tokyo Institute of Technology, Meguro, Tokyo, Japan

e-mail: sekijima@c.titech.ac.jp

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_55

759

and materials informatics is that the target values can be accurately calculated by quantum chemical calculations in material science [8, 9]. Quantum chemical calculations are the main method of approximation of physical properties, which are close to the experimental values. However, it is difficult to compute a large number of compounds comprehensively, including virtual compounds, because the amount of computation may increase significantly depending on the compound. Therefore, there has been a recent increase in research on materials screening and property prediction using machine learning methods using data in the materials database [10–12].

In the development of solar cells, perovskite is a more potent material type than conventional silicon type in recent years [13]. Perovskite solar cells can be made at a lower cost than the silicon type and have the advantages of being thinner and smaller, thus being less restricted in their installation location. However, perovskite solar cells, which are currently considered to have high performance, contain lead element (Pb), which is toxic for humans and the environment. It is necessary to select materials with appropriate band gap for the development of high-performance solar cells because the performance of solar cells highly depends on the band gap.

It is not realistic to calculate all of them by quantum chemical calculations because considering all monoatomic and polyatomic ions, the number of candidate materials for perovskite solar cells is enormous, including hypothetical compounds. In addition, in order to explore new materials efficiently using machine learning, it is necessary to be able to predict the properties of compounds for which no experiments or quantum chemical calculations have yet been performed, such as band gap. In this study, we developed a machine learning model that can predict the band gap of unknown (out-of-dataset) perovskite compounds, which cannot be covered by quantum chemical calculations.

2 Materials and Methods

2.1 Datasets

This study used public dataset [12]. This dataset contains the electronic state of Pb-free halide double perovskite, $A_2B^{1+}B^{3+}X_6$. The elements of the halide double perovskite in the dataset are as follows: the A site is potassium (K), rubidium (Rb), and cesium (Cs); the B^{1+} site is copper (Cu), silver (Ag), indium (In), gold (Au), and thallium (Tl); the B^{3+} site is aluminum (Al), gallium (Ga), indium (In), arsenic (As), antimony (Sb), and bismuth (Bi); and the X site is chlorine (Cl), bromine (Br), and iodine (I).

The detail of the dataset including the feature and labels is shown in Table 1. These features except distance and cubic are calculated for A, B1, B2, and X site. Distance is calculated for A, B, and X site. Cubic is assigned only one per compound. The total number of features is 32 at first and is reduced to 20 based on

Table 1 Dataset

Label	Description
distance	Atomic distance between A (B) site and X site
eleneg	Electro negativity of each site
hoe	Highest occupied atomic energy of each site
ionenergy	Ionization energy of each site
luep	Lowest unoccupied energy of each site
rs	Radius of s orbital of each site
rp	Radius of p orbital of each site
rd	Radius of d orbital of each site
cubic	Space group
ind_gap	Band gap (eV)

Table 2 Selected features

Cubic	distance_a	distance_b1	distance_b2
eleneg_a	eleneg_b1	eleneg_b2	eleneg_x
hoe_a	hoe_b1	hoe_b2	luep_a
luep_b1	luep_b2	rs_a	rs_b1
rs_b2	rd_a	rd_b1	rd_b2

the correlation of the features. The correlation coefficient R is expressed as Eq. 1:

$$R_{pq} = \frac{\sum_{i=1}^n (p_i - \bar{p})(q_i - \bar{q})}{\sqrt{\sum_{i=1}^n (p_i - \bar{p})^2 \sum_{i=1}^n (q_i - \bar{q})^2}} \quad (1)$$

where n denotes the number of compounds and \bar{p} and \bar{q} denote the mean of p_i and q_i , i -th feature of compounds p and q .

Selected features are shown in Table 2. The machine learning models are constructed from these features. Here, B^{1+} site and B^{3+} site are called B1 (b1) and B2 (b2) sites, respectively.

2.2 Machine Learning Methods

Here we describe the proposed method of data partitioning for band gap prediction focusing on a single element. This method is called leave-one-element-out cross-validation.

When deciding which compounds to synthesize in the laboratory and which elements to include, materials informatics techniques, based on machine learning, can be used. It is used to efficiently determine which compounds and their constituent elements have the desired properties. If the physical properties of compounds and constituent elements that do not exist in our data can be accurately predicted, we may be able to discover new materials that have not been envisioned before. In order to search for new materials, the physical properties of compounds with elements not

found in the training data are predicted. Specifically, a dataset partitioning method for the substitution of a single element is proposed.

An example of data splitting is shown in Fig. 1. Here, the data partitioning procedure for predicting the band gap of potassium (K) at site A is described. In order to predict the band gap of K-containing compounds, when dividing the dataset into training and test data of A site, select the compounds containing K for the test data and the compounds containing rubidium (Rb) or cesium (Cs), which are the same A site, for the training data. In this case, the number of training data is 360 and the number of test data is 180, respectively. The same procedure is performed for B1 sites (five types), B2 sites (six types), and X sites (three types). The band gap prediction is performed for each site.

A total of 17 elements were predicted using this method. Since the proportion of training and test data for sites A, B1, B2, and X is different, the number of data for each site is shown in Table 3.

In this study, the band gap of compound is predicted using XGBoost [14] model, which is based on the decision tree. XGBoost is an implementation of gradient-boosted trees, where the prediction is a weighted average of the decision trees. Each tree is trained sequentially, with changing the weight of each data point. The weights of wrongly predicted data points increase in the training of the next tree.

The loss function of XGBoost is defined as Eq. 2:

$$Loss(\theta) = L(\theta) + \Omega(\theta) \quad (2)$$

where θ is the parameter of the model and L and Ω are the training loss and regularization loss, respectively. In this study, L is the mean squared error:

$$L(\theta) = \sum_i (y_i - \hat{y}_i)^2 \quad (3)$$

Fig. 1 An example of data partitioning. Potassium (K) is chosen for the test data

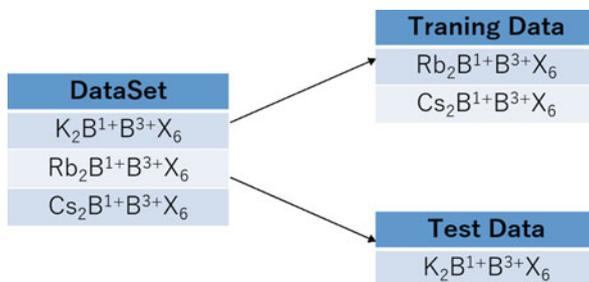


Table 3 The number of training and test data at each site

Site	Training	Test
A	360	180
B1	432	108
B2	450	90
X	360	180

where y_i and \hat{y}_i are the true and predicted labels of i -th compound, respectively. For the regularization term, a tree function $f(x)$ is defined in Eq. 4:

$$f(x) = w_{q(x)}, q \in R^T, q : R^d \rightarrow \{1, 2, \dots, T\} \quad (4)$$

Here w is the scores on leaves, q is a function from each data (x) to the corresponding leaf, and T is the number of leaves. In XGBoost, the complexity of a tree is defined as Eq. 5:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (5)$$

where γ and λ are the hyperparameters. The total complexity of the model is the summation of all trees.

2.3 Evaluation

In this study, coefficient of determination R^2 and root mean squared error ($RMSE$) are used as the evaluation metrics.

The definition of the coefficient of determination R^2 is described in Eq. 6:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (6)$$

Similarly, $RMSE$ is described in Eq. 7:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (7)$$

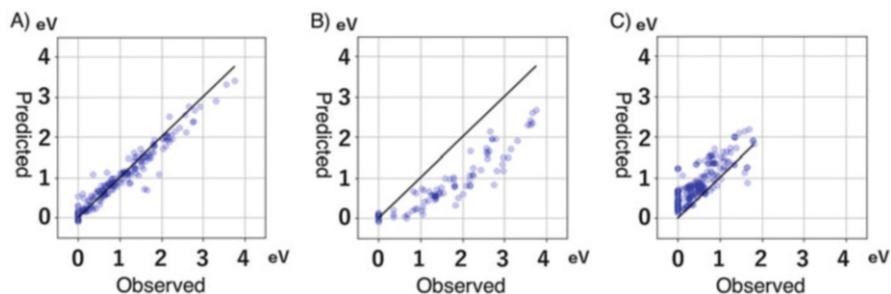
Here, y_i and \hat{y}_i are the true and predicted label of i -th compound, and \bar{y} is the mean of the labels, respectively.

3 Results

In this section, the results of the experiments conducted are presented. Table 4 shows the prediction results. Using reduced 20 features in the dataset, we compared the accuracy of the band gap prediction model constructed with one substitution in each element at R^2 and $RMSE$.

Table 4 Prediction results of each element using XGBoost

Site	Element	R^2	$RMSE$
A	K	0.935	0.212
	Rb	0.912	0.245
	Cs	0.865	0.288
B1	Cu	-2.22	0.755
	Ag	0.0581	0.564
	In	0.682	0.448
	Au	0.367	0.722
	Tl	0.436	0.655
B2	Al	0.289	0.919
	Ga	0.760	0.364
	As	0.643	0.354
	In	0.785	0.364
	Sb	0.919	0.173
	Bi	0.837	0.237
X	Cl	0.515	0.650
	Br	0.632	0.447
	I	0.118	0.450

**Fig. 2** Scatter plot of predicted and observed band gaps. (a) potassium (K), (b) aluminum (Al), and (c) iodine (I). Diagonal lines represent $y = x$

These results show that the accuracy at site A is higher than that at the other sites. As for other sites, it depends on the elements. Some elements are not accurate at all, some elements are highly accurate, and some of them are moderately accurate. To show this difference, we pick up some elements and compare them. The scatter plots of the predicted and observed potassium (K) at site A, which were relatively accurate, are shown in Fig. 2a. The scatter plots of aluminum (Al) at the B2 site and iodine (I) at the X site, which were relatively inaccurate, are shown in Fig. 2b and c. It can be seen that the band gap values are over(under)-evaluated as a trend of poor accuracy. A similar trend was observed for other elements to varying degrees.

4 Discussion

4.1 Number of Features

Figure 3 shows the accuracy against the change of the number of features used in the model. Figure 3a–d shows the R^2 for A, B1, B2, and X site, respectively.

For A site, the accuracy becomes stable with more than ten features for all elements. For other sites, almost the same tendency is applicable. However, there are several elements that the accuracy is not sufficient until 20 features, namely Cu, Al, and I. Especially, for Cu elements, the R^2 become lower when the number of features increases.

4.2 Estimation Bias

As shown in Fig. 3, the predicted band gap is underestimated for Al, and the band gap is overestimated for I for most of compounds. One reason lies in the dataset. Figure 4 shows the band gap values in the dataset. Figure 4a and b is for B2 site and X site, respectively. X-axis represents the compounds that contain each element, and Y-axis represents the band gap distribution (eV). All band gaps are higher or equal to 0.

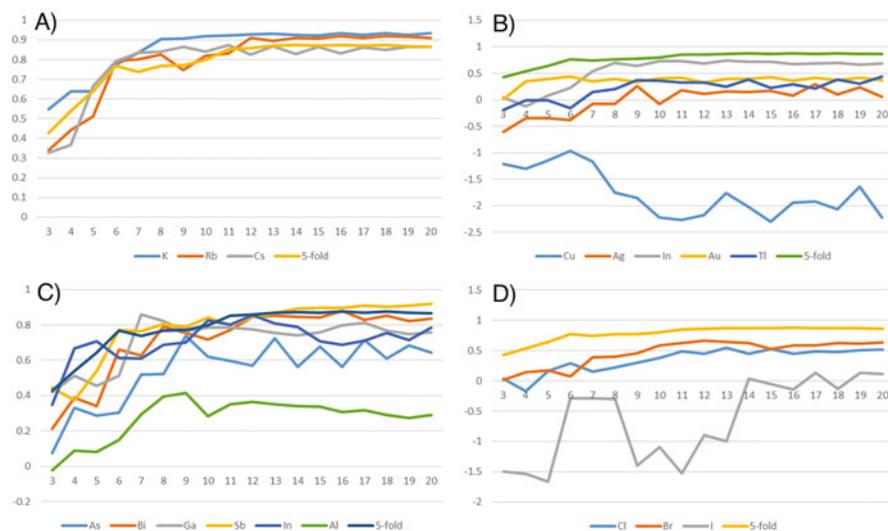


Fig. 3 The change of R^2 of the prediction against the number of the features. (a) A site, (b) B1 site, (c) B2 site, and (d) X site. Each line represents one element, and fivefold line represents ordinal fivefold cross-validation results, without considering leave-one-element-out cross-validation

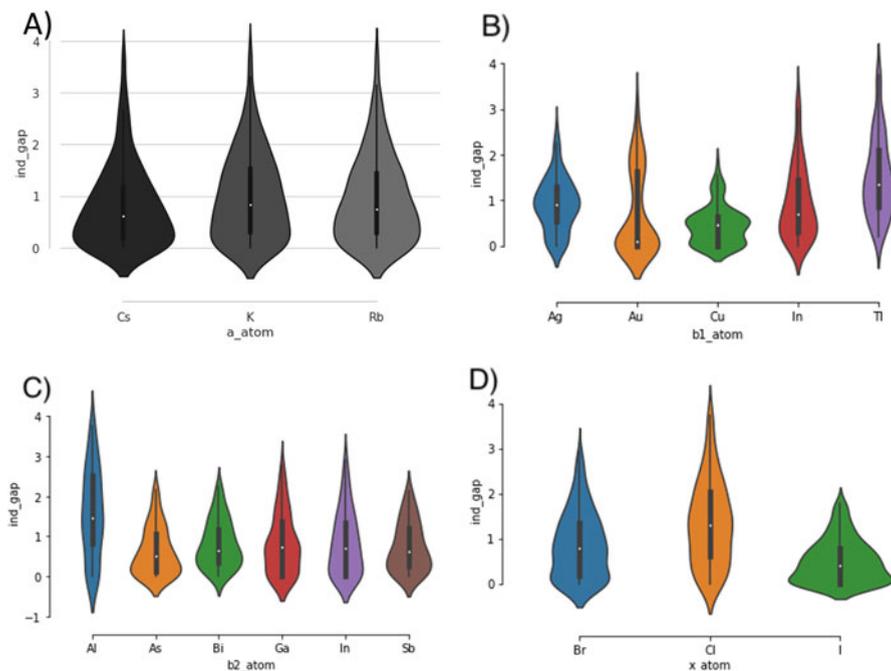


Fig. 4 Violin plot of the band gap in the dataset. (a) A site, (b) B1 site, (c) B2 site, and (d) X site

According to the plot, the distributions of band gap do not differ among A site elements and, however, differ among B2 and X site elements. Notably, the distribution of Al-containing compounds has broader distribution than others in B2 site, and the distribution of I-containing compounds has narrower distribution than others in X site. Thus, the distribution of the band gap may affect the prediction accuracy. Because a certain trend can be observed between the band gap distribution and the overestimation or underestimation, an error-correction method can be applied based on the current prediction model.

In this study, we constructed a band gap prediction model for Pb-free halide double perovskite compounds for single-element substitutions using machine learning. This study also focused on leave-one-element-out cross-validation, which divides compounds containing one element into test data and others into training data in cross-validation-like fashion. We also predicted the band gap for 17 elements in the data set.

The results of the experiments showed various trends, such as some of the selected elements had high accuracy and some had lower accuracy. One of the possible causes is the difference of the band gap distributions between training set and test set. This could lead a future new method to the prediction model that incorporates corrections for underestimation of overestimation.

This study only focused on halide double perovskites that are represented as $A_2B^{1+}B^{3+}X_6$. However, it is also possible to synthesize perovskites in which each site is partially substituted. Some examples are

- $(CH_3NH_3)Pb(Br_{1-x}Cl_x)_3$ [15]
- $Li_xLa_{(1-x)/3}NbO_3, (Li_{0.25}La_{0.25})_{1-x}Sr_{0.5x}NbO_3$ [16]
- $CsPb_{1-x}Sn_xIBr_2$ [17]

In order to predict the band gaps of these various perovskite compounds, which have not yet been experimentally or computed, the future challenge is to build a prediction model that takes into account perovskites with more substitutions.

Acknowledgments This work is partially supported by the Research Complex Program “Wellbeing Research Campus: Creating new values through technological and social innovation” from the Japan Science and Technology Agency (JST), the Platform Project for Supporting Drug Discovery and Life Science Research (Basis for Supporting Innovative Drug Discovery and Life Science Research (BINDS)) from AMED under Grant Number JP20am0101112, and the Japanese Society for the Promotion of Science (JSPS) KAKENHI Grant Number 20H00620.

References

1. Materials genome initiative. www.mgi.gov
2. S. Chiba, T. Ishida, K. Ikeda, M. Mochizuki, R. Teramoto, Y. Taguchi, M. Iwadate, H. Umeyama, C. Ramakrishnan, A.M. Thangakani, D. Velmurugan, M.M. Gromiha, T. Okuno, K. Kato, S. Minami, G. Chikenji, S.D. Suzuki, K. Yanagisawa, W.H. Shin, D. Kihara, K.Z. Yamamoto, Y. Moriwaki, N. Yasuo, R. Yoshino, S. Zozulya, P. Borysko, R. Stavniichuk, T. Honma, T. Hirokawa, Y. Akiyama, M. Sekijima, An iterative compound screening contest method for identifying target protein inhibitors using the tyrosine-protein kinase yes. *Sci. Rep.* **7**(1), 12038 (2017)
3. S. Chiba, M. Ohue, A. Gryniukova, P. Borysko, S. Zozulya, N. Yasuo, R. Yoshino, K. Ikeda, W.H. Shin, D. Kihara, M. Iwadate, H. Umeyama, T. Ichikawa, R. Teramoto, K.Y. Hsin, V. Gupta, H. Kitano, M. Sakamoto, A. Higuchi, N. Miura, K. Yura, M. Mochizuki, C. Ramakrishnan, A.M. Thangakani, D. Velmurugan, M.M. Gromiha, I. Nakane, N. Uchida, H. Hakariya, M. Tan, H.K. Nakamura, S.D. Suzuki, T. Ito, M. Kawatani, K. Kudoh, S. Takashina, K.Z. Yamamoto, Y. Moriwaki, K. Oda, D. Kobayashi, T. Okuno, S. Minami, G. Chikenji, P. Prathipati, C. Nagao, A. Mohsen, M. Ito, K. Mizuguchi, T. Honma, T. Ishida, T. Hirokawa, Y. Akiyama, M. Sekijima, A prospective compound screening contest identified broader inhibitors for sirtuin 1. *Scientific Reports* **9**(1), 1–12 (2019)
4. N. Yasuo, K. Watanabe, H. Hara, K. Rikimaru, M. Sekijima, Predicting strategies for lead optimization via learning to rank. *IPJSJ Trans. Bioinf.* **11**(0), 41–47 (2018). <https://doi.org/10.2197/ipsjtbio.11.41>
5. N. Yasuo, Y. Nakashima, M. Sekijima, CoDe-DTI: Collaborative deep learning-based drug-target interaction prediction, in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (IEEE, Piscataway, 2018), pp. 792–797
6. N. Arai, S. Yoshikawa, N. Yasuo, R. Yoshino, M. Sekijima, Compound property enhancement by virtual compound synthesis. *J. Bioinf. Comput. Biol. World Scientific Pub Co Pte Lt.* **16**(3), 1840016 (2018). <https://doi.org/10.1142/s0219720018400164>
7. N. Yasuo, M. Sekijima, Improved method of structure-based virtual screening via interaction-energy-based learning. *J. Chem. Inf. Model.* **59**(3), 1050–1061 (2019)

8. S. Kirklin, J.E. Saal, B. Meredig, A. Thompson, J.W. Doak, M. Aykol, S. Rühl, C. Wolverton, The open quantum materials database (OQMD): assessing the accuracy of DFT formation energies. *NPJ Comput. Mater.* **1**(1), 1–15 (2015)
9. J.E. Saal, S. Kirklin, M. Aykol, B. Meredig, C. Wolverton, Materials design and discovery with high-throughput density functional theory: the open quantum materials database (OQMD). *JOM* **65**(11), 1501–1509 (2013)
10. A. Jain, S.P. Ong, G. Hautier, W. Chen, W.D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder et al., Commentary: The materials project: A materials genome approach to accelerating materials innovation. *Apl Mater.* **1**(1), 011002 (2013)
11. S. Curtarolo, W. Setyawan, G.L. Hart, M. Jahnatek, R.V. Chepulskii, R.H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy et al., Aflo: an automatic framework for high-throughput materials discovery. *Comput. Mater. Sci.* **58**, 218–226 (2012)
12. J. Im, S. Lee, T.W. Ko, H.W. Kim, Y. Hyon, H. Chang, Identifying Pb-free perovskites for solar cells by machine learning. *NPJ Comput. Mater.* **5**(1), 1–8 (2019)
13. A. Kojima, K. Teshima, Y. Shirai, T. Miyasaka, Organometal halide perovskites as visible-light sensitizers for photovoltaic cells. *J. Am. Chem. Soc.* **131**(17), 6050–6051 (2009)
14. T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016), pp. 785–794
15. T. Zhang, M. Yang, E.E. Benson, Z. Li, J. van de Lagemaat, J.M. Luther, Y. Yan, K. Zhu, Y. Zhao, A facile solvothermal growth of single crystal mixed halide perovskite $\text{CH}_3\text{NH}_3\text{Pb}(\text{Br}_{1-x}\text{Cl}_x)_3$. *Chem. Commun.* **51**(37), 7820–7823 (2015)
16. Y. Kawakami, H. Ikuta, M. Wakihara, Ionic conduction of lithium for perovskite-type compounds, $\text{Li}_x\text{La}_{(1-x)/3}\text{NbO}_3$ and $(\text{Li}_{0.25}\text{La}_{0.25})_{1-x}\text{Sr}_{0.5x}\text{NbO}_3$. *J. Solid State Electrochem.* **2**(4), 206–210 (1998)
17. N. Li, Z. Zhu, J. Li, A.K.Y. Jen, L. Wang, Inorganic CsPbI_3 - XSnXBr_2 for efficient wide-bandgap perovskite solar cells. *Adv. Energy Mater.* **8**(22), 1800525 (2018)

Interpretation of ResNet by Visualization of the Preferred Stimulus in Receptive Fields



Genta Kobayashi and Hayaru Shouno

1 Introduction

In this decade, deep convolutional neural networks (DCNNs) have been used in many areas such as image processing, audio signal processing, language processing, and so on. Especially, in image classification task, DCNN showed higher performance rather than that of the previous works in the field of computer vision [10]. DCNN is a model in which the expressive power of features is greatly improved by deepening the hidden layer of the convolutional neural network (CNN). Characteristics of CNN are built to hierarchically stack convolutional layers and pooling layers. Both architectures are determined based on simple cells and complex cells that are the visual cortex of mammals [3]. CNN are added constraints from a biological point of view, e.g., weight sharing and sparse activation. LeCun et al. [8] propose a model of CNN called LeNet-5 for the classification task of digit images and apply the backpropagation algorithm of the gradient learning method to the model. Krizhevsky et al. [7] show the effectiveness DCNN on the natural image classification task. In the wake of their achievements, many researchers proposed various deep models [13, 15]. He et al. [4] also proposed a DCNN model called residual network (ResNet) that has skip connections for by-passing the layers. The ResNet improves the performance of the visual classification task drastically.

The success of DCNNs accelerated the need of understanding them from multiple angles. From the viewpoint of neuroscience, Yamins et al. experimentally showed the similarity between the visual cortex of the primate and a DCNN trained for classification task [17]. On the other hand, from an engineering viewpoint, the mainstream method of understanding DCNN is based on visualization of the inner

G. Kobayashi · H. Shouno (✉)

The University of Electro-Communications, Chofu, Tokyo, Japan
e-mail: genta-kobayashi@uec.ac.jp; shouno@uec.ac.jp

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_56

769

expression of DCNNs using the gradient backward projection [11, 12, 14]. These methods use the differentiability of the function of DCNNs in the task.

The basic structure of the DCNNs is based on the inspiration from the biological viewpoint [3]; however, non-biological improvements, which have been proposed in these years, increase the interpretation difficulties. For instance, ResNet is an improved model so that the gradient-based learning methods work well. To understand ResNet, Liao, and Poggio study the relation between a model of ResNet and the visual cortex [9]. They use the model of ResNet which is similar to the recurrent neural networks that had a feedback connection. The study shows the relationship between a model of ResNet and recurrent neural network, and then between the ventral stream and the model stacked recurrent neural network. However the model is added a strong constraint and is not commonly used.

In this research, in order to understand ResNet, we focus it from the viewpoint of the development of the preferred stimulus in receptive fields under the visual scene classification task with ImageNet [1, 10]. The receptive field is a basic concept of the visual cortex system. Roughly speaking, it means the part of the visual input area in which a neuron is able to respond. The preferred stimuli make the strong response of the neuron. We try to use the idea of the preferred stimulus in the receptive field to reveal properties of the ResNet.

2 Methods

2.1 Residual Network

He et al. proposed the concept of residual network (ResNet) and showed several models of ResNet, e.g., ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152 [4]. ResNet contains characteristic architecture called “skip connection” or “residual connection.” The concept of the residual connection is to divide the mapping function into linear and non-linear parts explicitly. Let an input vector as \mathbf{x} , the output vector as \mathbf{y} , and non-linear part of mapping function as $F(\cdot)$. Then skip connection is represented as

$$\mathbf{y} = \mathbf{x} + F(\mathbf{x}). \quad (1)$$

When the dimensions of \mathbf{x} and $F(\mathbf{x})$ are different, \mathbf{x} is mapped to sum them by a mapping function. The original ResNets introduce a down-sampling block which contains a convolutional layer at some skip connections. Figure 1 shows the schematic diagram of the components of the ResNet called residual block. In order to treat skip connection in the residual block, we introduce pseudo-feature maps for the identical part of Eq. (1). In the figure, each rectangle shows the feature map, the fixed arrows show the connectivity with trainable weights, and the dashed ones show the connectivity with a fixed weight. We also introduce PlainNet as the

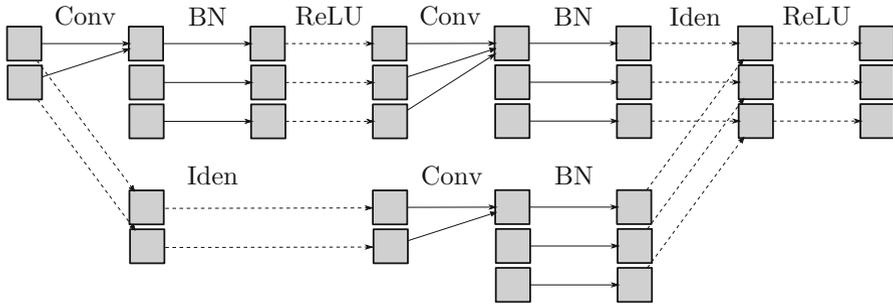


Fig. 1 Schematic diagram of the ResNet34: Each rectangle represents the feature map. The fixed arrows show the connectivity with trainable, and the dashed ones show the connectivity with a fixed weight. Conv, convolutional layer; BN, batch normalization; ReLU, ReLU function as $\max(x, 0)$; Iden, identity function

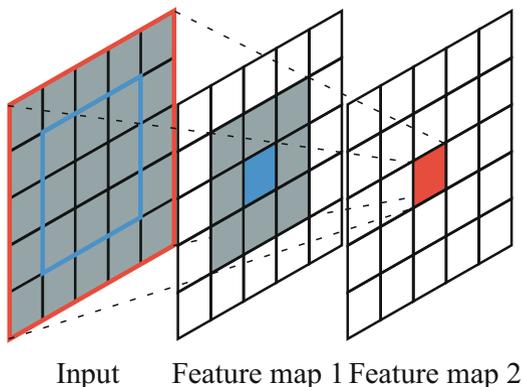
model excluding all the skip connections for comparison. We use ResNet34 and PlainNet34 for our experiment since the ResNet34 shows higher performance rather than those of the other ResNet models and previous DCNNs in our preliminary experiments.

2.2 Receptive Field

In the context of biological visual systems, the receptive field is the area on the retina to which a neuron has responded. It is considered that the receptive field contains the center and the surround area. Hubel and Wiesel shows almost all the receptive fields in the early visual cortex are very small [5] and they become large as the hierarchy deepens. Their work inspires the neocognitron [3], which is one of the origins of the DCNN, and influences many image recognition researches.

In the context of CNN, each neuron has the receptive field and also has preferred stimuli that are a part of the patch in the input image. Figure 2 shows an overview of the receptive field. The most right rectangle shows the feature map of the focused layer, and the middle and the left ones show the intermediate feature map and input, respectively. The feature map has neurons aligned with two-dimensional lattice. When we choose a neuron in the focused feature map, we can determine the connected area in the middle and the input. Thus, the preferred stimuli for the focused neuron appear in the red rectangle. Zeiler et al. [18] show samples of the preferred stimuli of DCNN and report the characteristic of each layer. Showing its sample is a simple method to understand trained features of CNN. We use the preferred stimulus to investigate the characteristic of neurons in this research. Let x be an image of $H \times W$, then the receptive field is a set of the spatial index. We can formally describe the receptive field image on the receptive field r of the image x as $x[r]$.

Fig. 2 Overview of receptive field. Each black border rectangle is a neuron. The area inside the blue border on input is the receptive field and corresponds to the blue neuron in feature map 1. The area inside the red border on input is the receptive field and corresponds to the red neuron in feature map 2



2.3 Visualization by Using Gradient

Many researchers use gradient base visualization methods to understand deep neural networks [2, 11, 12, 14]. First the work is activation maximization of Erhan et al. [2], and Simonyan et al. [12] apply it to DCNN. Activation maximization is to calculate an input that maximizes the activation of the neuron as an optimization problem. Let θ denote parameters of neural network, and let $f(\theta, \mathbf{x})$ be the activation of a neuron on a given input \mathbf{x} . Assuming a fixed θ , the method is represented as

$$\mathbf{x}^* = \arg \max_{\mathbf{x}} \{f(\theta, \mathbf{x}) - \lambda \|\mathbf{x}\|_2\}. \quad (2)$$

Since the solution we are interested in is a direction of input space, we add L_2 norm constraint and a regularization parameter λ . In general, this method is solved by gradient ascent of iterative methods because this is a non-convex optimization problem. This method can be applied to any differentiable models, but the resulting solution may be a boring local solution.

3 Experiment and Results

3.1 Training ResNets

We train ResNet34 and PlainNet34 with ImageNet dataset in the manner of He et al. [4] and Szegedy et al. [15]. The images in ImageNet have three color channels and are whitening with the channels. We apply the stochastic gradient descent method with an initial learning rate of 0.01 and a momentum of 0.9 and use a weight decay of 10^{-4} . The learning rate is divided by 10 every 30 epochs. The total training epoch is 90 with mini-batch size 256. In the training, the input images of 224×224 size

are randomly resized by an area scale between 8% and 100% and whose aspect ratio is chosen randomly between 3/4 and 4/3.

3.2 Visualization Filters

Teramoto and Shouno propose a visualization method for the preferred stimulus as a convolution filter in the second layer of VGG [16]. Let W_{pqij}^l be the convolutional weight connected from channel q in layer l to channel p in layer $l + 1$, and let i and j be the spatial indices. Then, the method is to use the weight \tilde{W}_p^2 as the p -th filter in the second layer. The weight \tilde{W}_p^2 is represented as

$$\tilde{W}_{pqij}^2 = \sum_k W_{pk\tilde{i}_{pk}\tilde{j}_{pk}}^2 W_{kqij}^1 \tag{3}$$

where $(\tilde{i}_{pk}, \tilde{j}_{pk}) = \arg \max_{i', j'} |W_{pk i' j'}^2|$. In general, this method is an approximated visualization for filters in higher layers because CNNs have a non-linear function between convolutional layers. We call the filter as ‘‘virtual filter’’ and apply this method to the second down-sampling layer in ResNet34. Figure 3 shows the virtual filters and the first filters in ResNet34. Looking at the coupling coefficients of the filters in Fig. 3, it can be seen that the coupling to similar filters is stronger. ResNet with a skip structure also acquires features similar to the column structure which is a biological finding.

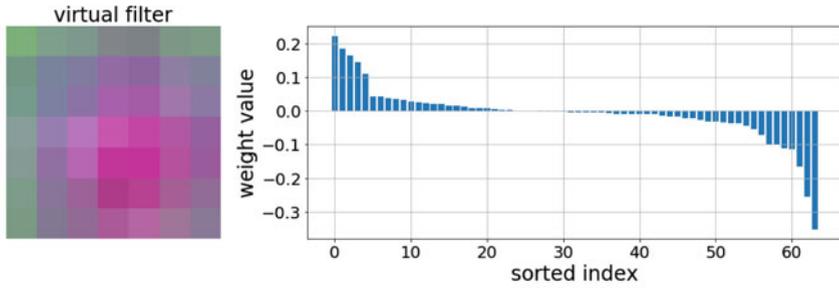
3.3 Analysis of the Preferred Stimulus in Receptive Fields

We focus on the preferred stimuli, which activate a neuron in the ResNets with strongly positive stimuli, in the input dataset. In order to find the preferred stimuli, we feed validation images X of ImageNet to DCNNs at first. After that, in each layer, we align the stimulus with a descending order of activation value. Let r_i be the receptive field of neuron i , and let $f_i(x[r_i])$ be the activation value of neuron i on a given receptive field image. Now, we can describe the mean preferred stimulus image on positive validation images X^+ as

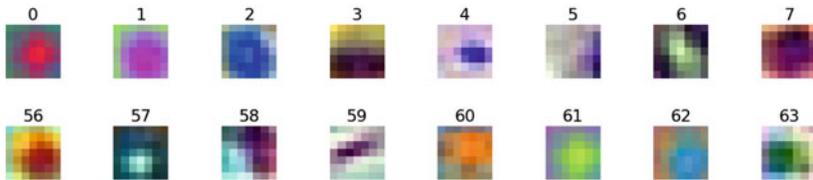
$$\bar{x}^i = \frac{1}{N} \sum_{x \in X^+} x[r_i]. \tag{4}$$

The positive validation images are validation images which the neuron activates positive and the images are represented by

$$X^+ = \{x \in X \mid f_i(x[r_i]) > 0\}. \tag{5}$$



(a)



(b)

Fig. 3 Visualization a filter of down-sampling shown at the bottom conv. layer of Fig. 1 in ResNet34. **(a)** Virtual filter \tilde{W}_2^2 and sorted the weight values. The right graph shows the values of weight $W_{2k\tilde{i}_{2k}\tilde{j}_{2k}}^2$, and sorted index of x-axis is an index k sorted in descending order. **(b)** Filters W^1 sorted by weight $W_{2k\tilde{i}_{2k}\tilde{j}_{2k}}^2$. The number above the image is a sorted index corresponding to **(a)**

We show a few examples of the top 16 at some neurons in Figs. 4 and 6, and the convolutional filter and the mean preferred stimulus images correspond to the neurons in Figs. 5 and 7. We find that DCNNs prefer a variety features as higher layers from the sample of the preferred stimuli. At first glance, Figs. 4c and 6c appear to be an inconsistent sample, but there are central features from Figs. 5d and 7d. We find that the characteristics of the same channel are similar in different layers due to the skip connection of the ResNet. We can see that the mean preferred stimulus images can only find the broad tendencies, but it is difficult to find the detailed properties of the neuron.

3.4 Visualization Using Maximization Method

We apply activation maximization method [2, 12] to ResNet34 and show the results for the neuron and the channel in the layer in Figs. 8 and 9. Optimizing for the neuron is to maximize the activation of the center neuron in a feature map, and optimizing for the channel is to maximize the average of the activation of a channel.

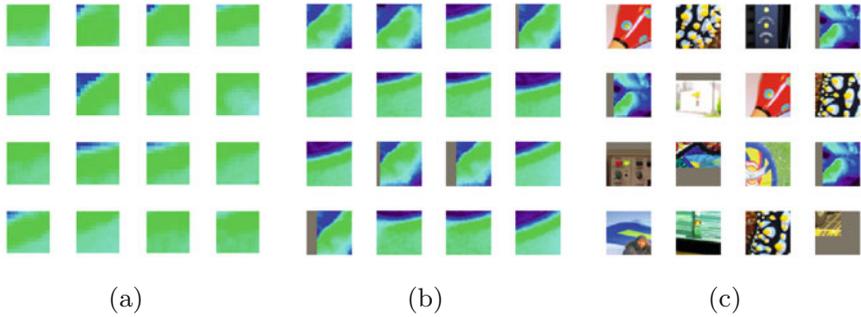


Fig. 4 Samples of the top 16 preferred stimulus images in ResNet34. (a) Channel 18 in first max-pooling layer. The receptive field size is 11×11 . (b) Channel 18 in conv. layer in layer 3. The receptive field size is 27×27 . (c) Channel 18 in conv. layer in layer 7. The receptive field size is 59×59

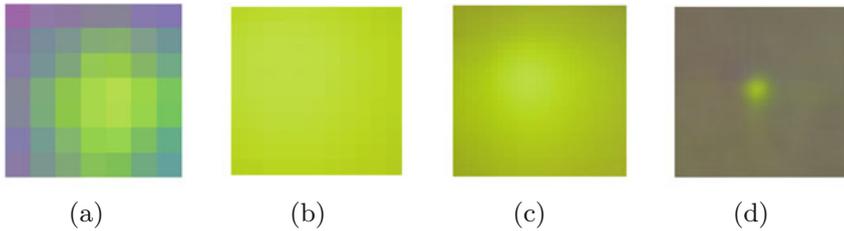


Fig. 5 First convolutional filter and mean preferred stimulus images in ResNet34. (a) First conv. filter of channel 18. (b) Mean preferred stimulus image of channel 18 in first max-pooling layer. (c) Mean preferred stimulus image of channel 18 in conv. layer in layer 3. (d) Mean preferred stimulus image of channel 18 in conv. layer in layer 7

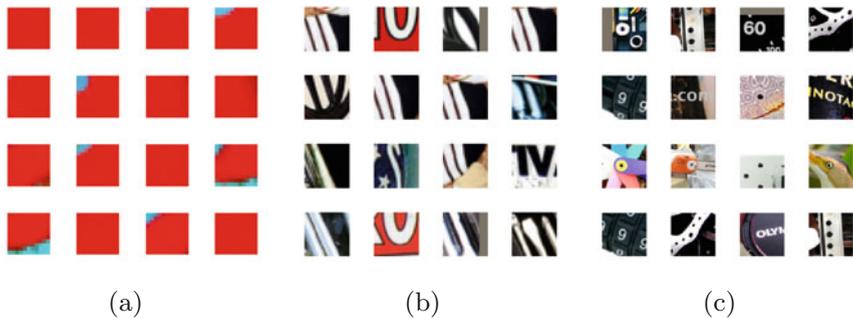


Fig. 6 Samples of the top 16 preferred stimulus images in PlainNet34. (a) Channel 19 in first max-pooling layer. (b) Channel 19 in conv. layer in layer 3. (c) Channel 19 in conv. layer in layer 7

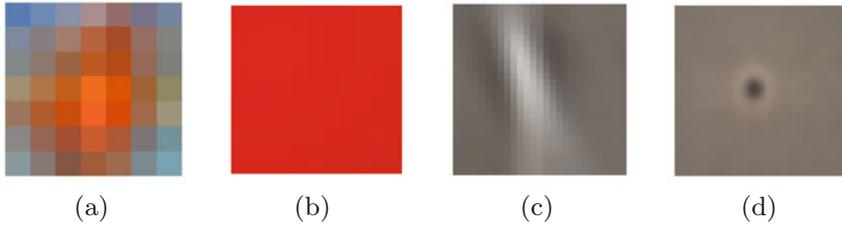


Fig. 7 First convolutional filter and mean preferred stimulus images in PlainNet34. (a) First conv. filter of channel 19. (b) Mean preferred stimulus image of channel 19 in first max-pooling layer. (c) Mean preferred stimulus image of channel 19 in conv. layer in layer 3. (d) Mean preferred stimulus image of channel 19 in conv. layer in layer 7

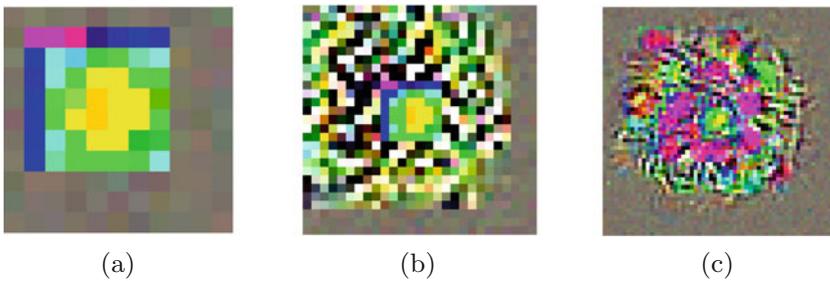


Fig. 8 Examples of visualizations by activation maximization for the neuron in ResNet34. (a) One optimal input of channel 18 in first max-pooling layer. (b) One optimal input of channel 18 in conv. layer in layer 3. (c) One optimal input of channel 18 in conv. layer in layer 7

We optimize the input by Adam optimizer [6] with a learning rate of 0.1 and a weight decay of 10^{-6} . In addition, we initialize the inputs from a zero image and iterate until 31 times.

From the comparison of Figs. 5 and 8, we can see that the results for optimizing for the neuron are similar to the results of the mean preferred stimulus images. The visualization at higher layers reveals the detailed properties for activation maximization but only simple trends for mean preferred stimulus images. Especially, visualizing by activation maximization for the channel is a good-looking visualization of the neuron, but the results vary according to various experimental conditions.

3.5 Inactive Neurons

For validation dataset images, we find that some channels in the first max-pooling layer have no output activation values; in other words output zeros value because of ReLU activation function. We call the channel as “inactive neuron.” In addition,

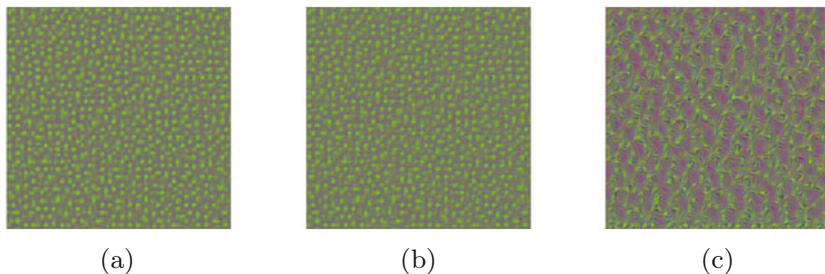


Fig. 9 Examples of visualizations by activation maximization for the channel in ResNet34. The image size is 224×224 . **(a)** One optimal input of channel 18 in first max-pooling layer. **(b)** One optimal input of channel 18 in conv. layer in layer 3. **(c)** One optimal input of channel 18 in conv. layer in layer 7

Table 1 Count of the inactive neurons and effect of the inactive neuron in first max-pooling layer for validation dataset in ResNet34 and PlainNet34

Model	# of inactive neurons	ΔL	ΔL_{rnd}
ResNet34	13	1.26962e+0	2.40560e-2
PlainNet34	2	-1.66893e-6	-8.34465e-7

we find that ResNet34 appears more as inactive neurons rather than that of the PlainNet34 from Table 1.

To investigate the effect of the inactivate neuron on the classification, we perform two classification experiments that add noise to the inactive neurons. The one is to add noise to all inactive neurons, and the second is to add noise to one inactive neuron selected randomly every mini-batch. We apply noise $\epsilon = \max(x, 0)$ where $x \sim \mathcal{N}(0, 1)$ to each spatial dimension of the inactive neuron. Table 1 shows the results: ΔL means the value from all noised validation loss minus validation loss, and ΔL_{rnd} means the value from randomly noised validation loss minus validation loss. We can see that the inactive neuron of ResNet34 effects classification task because both ΔL and ΔL_{rnd} of ResNet34 are positive and bigger than that of PlainNet34.

4 Conclusion

We perform the analysis by using the preferred stimulus and activation maximization to ResNets. Using both methods, we can find that ResNet has orientation-selective neurons and double-opponent color neurons. Both methods are able to characterize the lower layers well, but it is harder to use the analysis for the higher layers. We find that there are inactive neurons for the classification task in ResNet34. We speculate that this phenomenon is due to channel sharing by skip connections.

One hypothesis is that some channels are used for features that are not similar to the features of the first convolutional layers. In future works, we need to consider methods that can perform the analysis to the higher layers and examine the evidence to support our hypothesis.

References

1. J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in *CVPR09* (2009)
2. D. Erhan, Y. Bengio, A. Courville, P. Vincent, Visualizing higher-layer features of a deep network. *Univ. Montreal* **1341**(3), 1 (2009)
3. K. Fukushima, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* **36**(4), 193–202 (1980). <https://doi.org/10.1007/BF00344251>
4. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778. <https://doi.org/10.1109/CVPR.2016.90>
5. D.H. Hubel, T.N. Wiesel, Receptive fields of single neurones in the cat's striate cortex. *J. Physiol.* **148**(3), 574–591 (1959)
6. D.P. Kingma, J. Ba, Adam: A method for stochastic optimization (2014). Preprint. arXiv:1412.6980
7. A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in *Advances in Neural Information Processing Systems*, ed. by F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger, vol. 25 (Curran Associates, Northampton, 2012), pp. 1097–1105. <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
8. Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition. *Proc. IEEE* **86**, 2278–2324 (1998)
9. Q. Liao, T. Poggio, Bridging the gaps between residual learning, recurrent neural networks and visual cortex (2016). Preprint. arXiv:1604.03640
10. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge. *Int. J. Comput. Vision (IJCV)* **115**(3), 211–252 (2015). <https://doi.org/10.1007/s11263-015-0816-y>
11. R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in *2017 IEEE International Conference on Computer Vision (ICCV)* (IEEE, Piscataway, 2017), pp. 618–626
12. K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps (2013). Cornell University repository for scholarly material. <https://arxiv.org/abs/1312.6034>; submitted on 20 Dec 2013 (v1), last revised 19 Apr 2014 (this version, v2)] last accessed June 15, 2021
13. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition (2014). Preprint. arXiv:1409.1556
14. J.T. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for simplicity: The all convolutional net (2014). Preprint. arXiv:1412.6806
15. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 1–9
16. T. Teramoto, H. Shouno, A study of inner feature continuity of the VGG model, in *IEICE Technical Report. IEICE* (2019), pp. 239–244

17. D.L.K. Yamins, H. Hong, C.F. Cadieu, E.A. Solomon, D. Seibert, J.J. DiCarlo, Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proc. Natl. Acad. Sci.* **111**(23), 8619–8624 (2014). <https://doi.org/10.1073/pnas.1403112111>. <https://www.pnas.org/content/111/23/8619>
18. M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in *European Conference on Computer Vision* (Springer, Cham, 2014), pp. 818–833

Bayesian Sparse Covariance Structure Analysis for Correlated Count Data



Sho Ichigozaki, Takahiro Kawashima, and Hayaru Shouno

1 Introduction

Revealing correlations of data is the simplest way to analyze relationships of given samples. However, the simple way has a lot of unignorable problems such as noise robustness, interpretability, and treatments of discrete data. Specifically, using simple correlations only, large input dimension causes difficulty of finding significant relationships because the reasonable threshold selection is not trivial. In order to find the essential relationships between variables, the sparse modeling, such as LASSO (least absolute shrinkage and selection operator) [8], is focused in these decades [4]. Graphical Lasso [2] is a representative method which achieves sparse covariance structure analysis. Since an inverse covariance matrix corresponds to a partial correlation matrix with appropriate scaling, we can discover robust and essential relationships of data by sparse covariance structure analysis. However, because of the assumption of a Gaussian Graphical Model (GGM) for observed data, Graphical Lasso can not treat count data.

In order to overcome this limitation of Graphical Lasso, we propose a hierarchical Bayesian model for Poisson-distributed observations with sparse covariance structure. In our model, the latent variables, which indicate “potential risks” of events, follow a Bayesian Graphical Lasso (BGL) [9], and the occurrences of events

S. Ichigozaki

The University of Electro-Communications, Chofu, Tokyo, Japan

National Police Agency Info-Communications Bureau, Chiyoda-ku, Tokyo, Japan

e-mail: ichigozaki.show@uec.ac.jp

T. Kawashima · H. Shouno (✉)

The University of Electro-Communications, Chofu, Tokyo, Japan

e-mail: shouno@uec.ac.jp

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_57

781

follow the homogeneous Poisson processes. We apply the proposed model to spatial crime data analysis and investigate the effectiveness with numerical experiments.

1.1 Graphical Lasso

Graphical Lasso [2] is a well-known and powerful model for sparse covariance structure analysis of GGM. Let a zero-meaned data matrix $Y = (y_1, \dots, y_T) \in \mathbb{R}^{T \times A}$ follow a Gaussian distribution independently given a precision matrix $\Omega \in \mathbb{R}^{A \times A}$ denoting each component as ω_{ij} in $(i, j = 1, \dots, A)$:

$$p(Y|\Omega) = \prod_{t=1}^T N(y_t|\mathbf{0}, \Omega^{-1}). \tag{1}$$

The Graphical Lasso optimizes the following objective which consists of the term from maximizing likelihood estimator of Ω under the L_1 penalty:

$$\Omega = \arg \max_{\Omega' \in \mathbf{M}^+} \log(\det(\Omega')) - \text{tr}(Y^T Y \Omega') - \lambda \|\Omega'\|_1, \tag{2}$$

where \mathbf{M}^+ indicates the set of all $A \times A$ positive semi-definite matrices and λ is a regularization parameter. Here, the L_1 norm is defined by $\|\Omega\|_1 = \sum_{ij} |\omega_{ij}|$. Note that maximizing the objective (2) of Graphical Lasso is equivalent to obtain the maximum a posteriori (MAP) estimator of the following model, which is the combination of the Laplace distributions of the off-diagonal components of Ω denoting as $\text{DE}(\omega_{ij} \mid \lambda)$ of the form $p(\omega_{ij}) = \lambda/2 \exp(-\lambda|\omega_{ij}|)$ and the exponential distribution of the diagonal components $\text{Exp}(\omega_{ii} \mid \lambda/2)$ of the form $p(\omega_{ii}) = \lambda/2 \exp(-\lambda\omega_{ii}/2)$:

$$p(\Omega \mid \lambda) = C^{-1} \prod_{i < j} \{\text{DE}(\omega_{ij} \mid \lambda)\} \prod_{i=1}^A \text{Exp}\left(\omega_{ii} \mid \frac{\lambda}{2}\right) \mathbf{1}_{\Omega \in \mathbf{M}^+}, \tag{3}$$

where C^{-1} is a normalizing term and $\mathbf{1}_{\Omega \in \mathbf{M}^+}$ is an indicator function defined by

$$\mathbf{1}_{\Omega \in \mathbf{M}^+} = \begin{cases} 1 & (\Omega \in \mathbf{M}^+) \\ 0 & (\text{otherwise}). \end{cases} \tag{4}$$

1.2 Bayesian Graphical Lasso

Bayesian Graphical Lasso (BGL) [9] realizes a fully Bayesian treatment of Graphical Lasso. Using the fact that the double exponential distributions can be represented as a mixture of Gaussian and exponential distributions, (3) is modified to

$$p(\boldsymbol{\Omega}|\boldsymbol{\tau}, \lambda) = C_{\boldsymbol{\tau}}^{-1} \prod_{i < j} N(\omega_{ij}|0, \tau_{ij}) \prod_{i=1}^A \text{Exp}\left(\omega_{ii} \mid \frac{\lambda}{2}\right) \mathbf{1}_{\boldsymbol{\Omega} \in M^+}, \tag{5}$$

$$p(\boldsymbol{\tau}|\lambda) \propto C_{\boldsymbol{\tau}} \prod_{i < j} \frac{\lambda^2}{2} \exp\left(-\frac{\lambda^2}{2}\tau_{ij}\right), \tag{6}$$

where $\boldsymbol{\omega} = \{\omega_{ij}\}_{i \leq j}$ denotes the vector of the upper off-diagonal and diagonal entries of $\|\boldsymbol{\Omega}\|$ and $\boldsymbol{\tau} = \{\tau_{ij}\}_{i < j}$ is the latent scale parameters. From the decomposition of the double exponential distributions, we derive a data-augmented block Gibbs sampling algorithm for the BGL model(5).

1.3 The Poisson Process and Crime Data

A Poisson distribution gives probability masses of the numbers of occurred events per unit time. More precisely, if the time intervals of events' occurrences follow identical exponential distributions, the numbers of events per unit time follow identical Poisson distributions. This model is called a homogeneous Poisson process. The Poisson process is a simple but reasonable model to represent occurrences for rare events. Common examples of Poisson processes are customers calling a help [10], radioactive decay in atoms [7], crime occurrence [6], anomaly detection [5], and so on.

If data follow a multivariate point process, it is significant to understand the inter-variate relationships of data. When the data consists of continuous values, we can evaluate the structure of variables by calculating correlations or applying Graphical Lasso. However, for count data, the correlation matrix will lead to biased evaluation for the structure. Hence, we introduce a novel Bayesian framework of sparse covariance structure analysis for count data.

The remainder of this paper is organized as follows. Section 2 gives the concrete formulation of the proposed model. In Sect. 3, we show the effectiveness of our model through experiments with spatial crime data, and Sect. 4 shows the discussion of our model. Finally, Sect. 5 is devoted to a summary of this study.

2 The Proposed Method

2.1 Our Model

Figure 1 shows the graphical model of the proposed model. At first, we define the set of non-negative integers $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. When we obtain A -dimensional sequential data for T time steps, we assume that the elements of count data matrix $\mathbf{Y} \in \mathbb{N}_0^{T \times A}$ follow conditionally independent Poisson distributions:

$$P(\mathbf{Y}|\boldsymbol{\mu}, \mathbf{Z}) = \prod_{i=1}^A \prod_{t=1}^T \text{Poisson}(y_{ti} | \exp(\eta(\mu_i, z_{ti}))), \tag{7}$$

$$\eta(\mu_i, z_{ti}) = \mu_i + z_{ti}, \tag{8}$$

and we also give priors as

$$p(\boldsymbol{\mu}) = N(\boldsymbol{\mu}|\mathbf{0}, \sigma_\mu^2 \mathbf{I}), \tag{9}$$

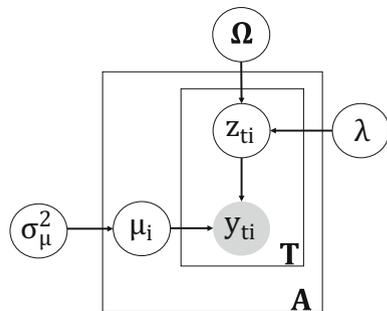
$$p(z_t|\boldsymbol{\Omega}) = N(z_t|\mathbf{0}, \boldsymbol{\Omega}^{-1}) \text{ for } t = 1, \dots, T, \tag{10}$$

$$p(\boldsymbol{\omega}|\lambda) \propto \prod_{i < j} \text{DE}(\omega_{ij}|\lambda) \prod_{i=1}^A \text{Exp}\left(\omega_{ii} \mid \frac{\lambda}{2}\right) \mathbf{1}_{\boldsymbol{\Omega} \in \mathbf{M}^+}, \tag{11}$$

$$p(\lambda) = \text{Gamma}(\lambda|a_\lambda, b_\lambda). \tag{12}$$

Here, we assume the linear predictor $\eta(\mu_i, z_{ti}) = \mu_i + z_{ti}$ as the *potential risk* of occurrence of events. Thus, μ_i indicates an averaged potential risk of i -th dimension, and z_{ti} represents dispersities from μ_i . Since z_t follows the BGL prior, we can extract the sparse and essential co-occurrence structures of count data. In addition, we discuss about the effects of choices of $p(\lambda)$ in Sect. 3. Therefore, the joint posterior distribution can be expressed as

Fig. 1 The graphical model of the proposed model



$$p(\mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Omega}, \lambda | \mathbf{Y}) \propto P(\mathbf{Y} | \boldsymbol{\mu}, \mathbf{Z}) p(\boldsymbol{\mu}) \left[\prod_{t=1}^T p(z_t | \boldsymbol{\Omega}) \right] p(\boldsymbol{\omega} | \lambda) p(\lambda). \tag{13}$$

2.2 Sampling Scheme

We evaluate the posterior (13) with a Markov chain Monte Carlo method which consists of two different sampling schemes. The first scheme is block Gibbs sampling for the parameters in the BGL, *i.e.*, $\boldsymbol{\Omega}$ and λ . The second one is about the parameters of the potential risks, that is, $\boldsymbol{\mu}$ and \mathbf{Z} . We adopt the Metropolis-Hastings scheme for correlated count data [1].

For example, we describe the Metropolis-Hastings sampling scheme from the fully conditional distribution of $\boldsymbol{\mu}$ here. When $\boldsymbol{\mu}$ tries to transition to new state $\boldsymbol{\mu}'$, we define the acceptance probability r for simplicity:

$$r = \min \left\{ 1, \frac{p(\boldsymbol{\mu}' | \mathbf{Y}, \mathbf{Z}, \boldsymbol{\Omega}, \lambda) p(\boldsymbol{\mu} | \boldsymbol{\mu}')}{p(\boldsymbol{\mu} | \mathbf{Y}, \mathbf{Z}, \boldsymbol{\Omega}, \lambda) p(\boldsymbol{\mu}' | \boldsymbol{\mu})} \right\} \tag{14}$$

The sampling procedure is as follows:

Step 1 Find optimal $\boldsymbol{\mu}$ about the fully conditional distribution

$$\hat{\boldsymbol{\mu}} = \arg \max_{\boldsymbol{\mu}} \log p(\boldsymbol{\mu} | \mathbf{Y}, \mathbf{Z}, \boldsymbol{\Omega}, \lambda) \tag{15}$$

by using the Newton-Raphson method.

Step 2 Sample a candidate state $\boldsymbol{\mu}'$ from proposal distribution $p(\boldsymbol{\mu}' | \hat{\boldsymbol{\mu}})$ defined as the multivariate t-distribution:

$$p(\boldsymbol{\mu}' | \hat{\boldsymbol{\mu}}) = \text{Multi} - t(\boldsymbol{\mu}' | \hat{\boldsymbol{\mu}}, \mathbf{H}_{\hat{\boldsymbol{\mu}}}^{-1}, \nu), \tag{16}$$

where $\mathbf{H}_{\boldsymbol{\mu}} \in \mathbf{R}^{A \times A}$ is the Hessian matrix of (15) and $\nu (> 0)$ is a user-defined hyperparameter which indicates the degree of freedom.

Step 3 Accept the candidate state $\boldsymbol{\mu}'$ with probability r .

Update the state $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu}'$ if accepted, and keep the state to be if rejected.

Sampling from $p(z_t | \mathbf{Y}, \mathbf{Z}_{\setminus t}, \boldsymbol{\mu}, \boldsymbol{\Omega}, \lambda)$ is also executed by the similar way to $\boldsymbol{\mu}$. In summary, we repeat sampling from the fully conditional distributions of $\boldsymbol{\Omega}$, λ , $\boldsymbol{\mu}$, and \mathbf{Z} , respectively, with Metropolis-Hastings within Gibbs sampler.

3 Synthetic Data Analyses

3.1 Synthetic Data

To assess the performance of our proposed model, we generate four synthetic datasets, whose size are $(A, T) = (10, 30), (50, 60), (100, 60), (200, 60)$. We fix $\mu_i = 0.2$ ($i = 1, \dots, A$) and generated $\mathbf{\Omega}$ as \mathbf{M}^+ with $\omega_{ii} = C_1$, $\omega_{i, i-A/2} = \omega_{i-A/2, i} = C_2$ and zero otherwise for the entire simulations, where C_1 and C_2 represent constant values. Given these true parameters, we sample \mathbf{Z} and the observed data matrix \mathbf{Y} from (7) and (10), respectively.

3.2 Analyzing Effects of Hyperparameter Selection

In our Metropolis-Hastings sampling scheme, we empirically find that an appropriate selection for the degree of freedom ν in proposal t-distribution gives a significant effect for the sampling efficiency. In our simulations, slightly small ν , such as $\nu = 5$, is better than bigger ν . Because a t-distribution with $\nu = 1$ is equivalent to Cauchy distribution and $\nu \rightarrow \infty$ is corresponding to a Gaussian distribution, our choice $\nu = 5$ means the intermediate form of them.

Next, we determine the parameters in priors. For a_λ , which is the hyperparameter of the regularization parameter λ , we adopt $a_\lambda = A$ for $A = 10, 50, 100$, and $a_\lambda = 0.01$ for $A = 200$. Moreover, a value $\sigma_\mu^2 = 0.05$ seems to be reasonable for the prior $p(\boldsymbol{\mu})$. Since $\boldsymbol{\mu}$ and \mathbf{Z} determine the parameters of the Poisson distribution within the exponential function in the proposed model, the absolute value of μ should be small.

3.3 Simulation Results

Here we show the estimation results of $\boldsymbol{\mu}$, \mathbf{Z} , and $\mathbf{\Omega}$ in the case that the size of data is $(A, T) = (50, 60)$. We adopt the hyperparameters $(a_\lambda, \sigma_\mu^2, \nu) = (A, 0.05, 3)$ and updated all elements at once both $\boldsymbol{\mu}$ and \mathbf{z}_t in the Metropolis-Hastings algorithms.

Figure 2 shows the result of $\boldsymbol{\mu}$. We show the solid line estimates of μ_i with the MAP estimator with 95% credible intervals. The dotted line represents the true values. It is assumed that small μ_i is difficult to estimate because the linear predictor $\eta(\mu_i, z_{ti})$ is exponentially transformed to be the parameter of Poisson distribution. However, we find the true values almost are held within the 95% credible intervals.

Figure 3 shows the MAP estimation results of \mathbf{Z} and their true values. Subfigure (c) shows the differences between MAP estimated and true \mathbf{Z} in absolute value. Figure 4 shows the estimation result of the value of $(z_{t1})_{t=1}^T$. Note that μ_1 is estimated to be 0.2 as shown in Fig. 2. The solid line is the MAP estimation results of $(z_{t1})_{t=1}^T$ with the 95% credible intervals, and the markers are their true values.

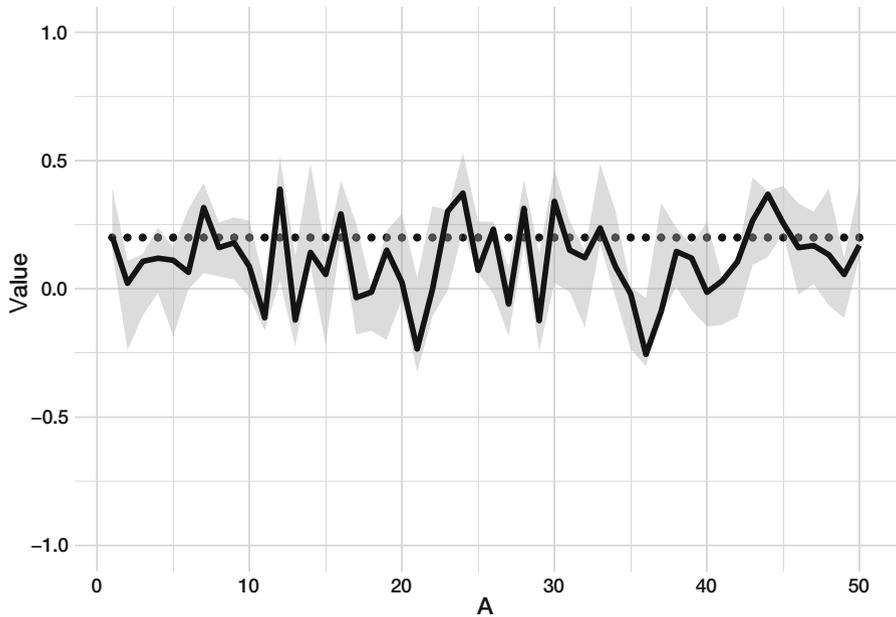


Fig. 2 The estimation result of μ with the MAP estimator with 95% credible intervals

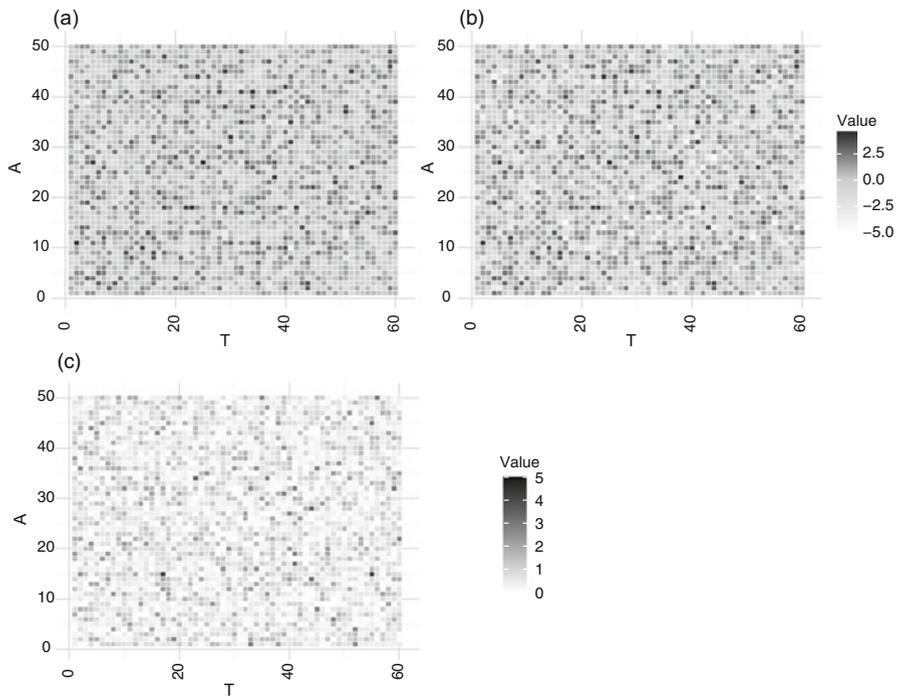


Fig. 3 (a) shows the MAP estimation results of Z , (b) is true value of Z , and (c) shows the difference between (a) and (b) in absolute value

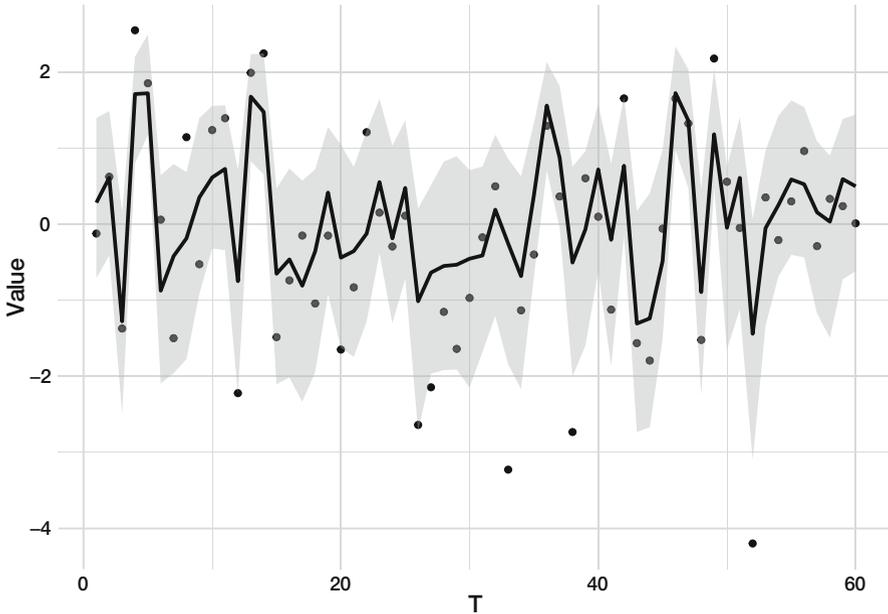


Fig. 4 The estimation result of $(z_{t1})_{t=1}^T$. The markers show the true values of and the solid line shows MAP estimated values with the 95% credible intervals

For the positive z_{t1} , the true values are within the credible intervals of the estimation result. On the other hand, the true values seem to be out of the 95% credible intervals for the negative z_{t1} . However, considering that the parameter of Poisson distribution becomes a small value, the estimation results are reasonable.

3.4 Evaluating Partial Correlations

We can calculate partial correlation matrix \mathbf{P} as

$$P_{ij} = -\frac{\omega_{ij}}{\sqrt{\omega_{ii}}\sqrt{\omega_{jj}}} \tag{17}$$

from the estimated precision matrix $\mathbf{\Omega}$. In Fig. 5, we show the MAP estimated \mathbf{P} for $(A, T) = (50, 60)$. We compare the estimated partial correlation coefficients and true one, which consists of a small number of non-zero components. We confirm the partial correlation coefficients that correspond to the non-zero components are relatively larger than the other components that correspond to the zero components in the true partial correlation coefficients. We also find that non-diagonal and non-

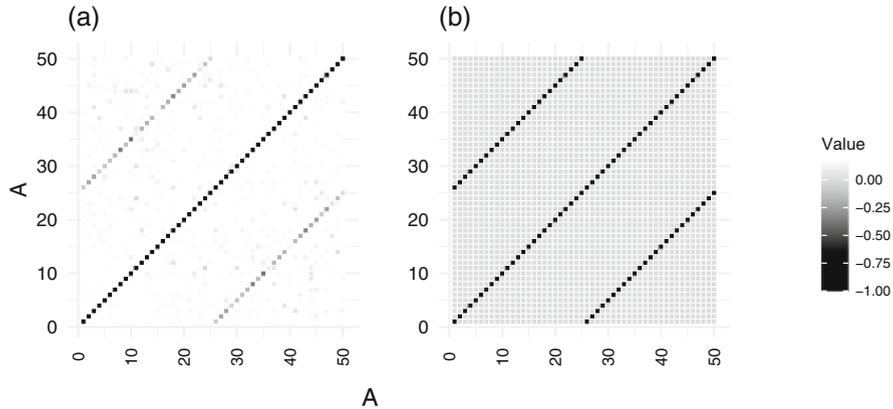


Fig. 5 (a) shows the estimated partial correlation coefficients of Ω and (b) shows the true partial correlation coefficients of Ω

zero estimates have shrinkage from their true values. This is affected by the Lasso-like priors of BGL.

4 Analysis of Crime Spots Data

4.1 Spatial Crime Data

As an application example of the proposed model, we employed the proposed model to spatial crime data which is obtained from [3]. The spatial crime data published by the National Institute of Justice (NIJ) contains criminal occurrences and their locations in the Portland City, Oregon, USA.

In the crime data, the latent variable μ_i represents the average potential risk of criminal occurrences of the i -th area. Also, \mathbf{Z} represents dispersities of the potential risks at each time point. Therefore, the interaction structure is captured by \mathbf{Z} .

For simplicity, we used the crimes that occurred in 2016 and extracted 60 areas. Furthermore, we aggregate the number of crimes per week into one time point. Hence, the size of the data matrix becomes \mathbf{Y} which is $(A, T) = (60, 52)$.

4.2 The Partial Correlation of Ω on Crime Data

The estimated Ω is the inverse of the sparse covariance matrix that contributes to the influence \mathbf{Z} between variables and temporal variation. Therefore, the partial

correlation calculated from Ω represents a sparse correlation of crime occurrence risk between areas.

A strong positive correlation between two areas means that the risk of the crime in one area intends to increase when the other area's risk increases.

4.3 Visualization of Partial Correlations

The partial correlation coefficients matrix is calculated from (17) with the estimated result of Ω . Figure 6 shows a visualized sparse partial correlation coefficients on the Portland City's map. The coefficients shown on the map are the top 2% of the whole in absolute value. The correlation of crime risks could lead to a visual understanding by expressing a strong correlation on the map.

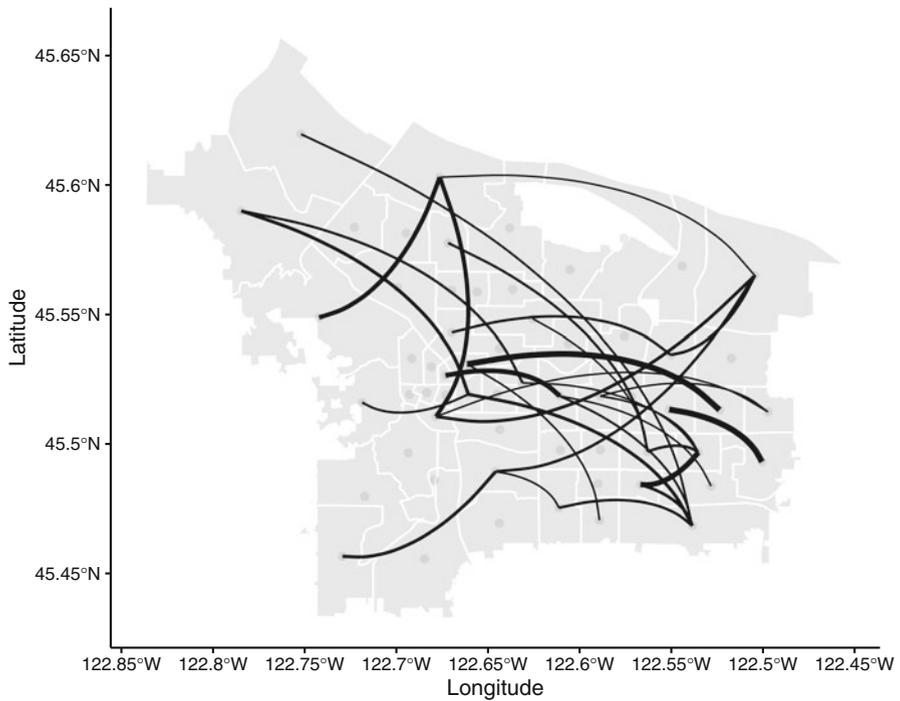


Fig. 6 The visualization of sparse partial correlation coefficients between areas on Portland City's map. The thickness of the curved black lines shows the magnitudes of the coefficients

5 Conclusion

The proposed model can estimate reasonable values of Ω and latent variables μ and Z . On the other hand, when the parameter is less than 1 in the Poisson process, the number of events tends to be 0, so that it is difficult to estimate the negative true value of the latent variable of the simulation data in the proposed model. One of the possible solutions that tackles this problem is to find an alternative transformation function which maps \mathbb{R} to $[0, \infty)$ for the linear predictor of Poisson distribution. For example, $x \mapsto \log(1 + \exp(x))$ is one of the candidates.

It is possible to find useful features by analyzing the estimation results of latent variables by the proposed model. We have been able to obtain sparse correlations between crime-risk areas, by applying the proposed model to crime data. As a future issue, because there are many samples related to time in the Poisson process in general, we would like to make our model possible to catch time series dependence.

References

1. S. Chib, I. Jeliazkov, Marginal likelihood from the metropolis–hastings output. *J. Am. Stat. Assoc.* **96**(453), 270–281 (2001)
2. J. Friedman, T. Hastie, R. Tibshirani, Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **9**, 432–441 (2008)
3. N. Homepage, Real-time crime forecasting challenge. <https://nij.ojp.gov/funding/real-time-crime-forecasting-challenge>. Last accessed 18 May 2020
4. Y. Igarashi, K. Nagata, T. Kuwatani, T. Omori, Y. Nakanishi-Ohno, M. Okada, Three levels of data-driven science. *J. Phys. Conf. Ser.* **699**, 012001 (2016). <https://doi.org/10.1088/1742-6596/699/1/012001>
5. A. Ihler, J. Hutchins, P. Smyth, Adaptive event detection with time-varying poisson processes, in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2006), pp. 207–216
6. D.W. Osgood, Poisson-based regression analysis of aggregate crime rates. *J. Quant. Criminol.* **16**(1), 21–43 (2000)
7. A. Sitek, A.M. Celler, Limitations of poisson statistics in describing radioactive decay. *Phys. Med.* **31**(8), 1105–1107 (2015)
8. R. Tibshirani, Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. B (Methodological)* **58**(1), 267–288 (1996). <http://www.jstor.org/stable/2346178>
9. H. Wang et al., Bayesian graphical lasso models and efficient posterior computation. *Bayesian Anal.* **7**(4), 867–886 (2012)
10. J. Weinberg, L.D. Brown, J.R. Stroud, Bayesian forecasting of an inhomogeneous poisson process with applications to call center data. *J. Am. Stat. Assoc.* **102**(480), 1185–1198 (2007)

Gaze Analysis of Modification Policy in Debugging an Embedded System



Takeru Baba, Erina Makihara, Hirotaka Yoneda, Kiyoshi Kiyokawa, and Keiko Ono

1 Introduction

Several investigations have reported that the demand for Internet of Things (IoT) developers is increasing with the rise in the size of the IoT market. To develop an IoT system, a broad knowledge of embedded system, network, and distributed data processing is required. Embedded system development including IoT requires not only the development of software but also that of hardware (e.g., micro-controllers and circuits), thereby increasing the number of domains where system errors can occur. In software development, a developer can check the system state in real time by outputting exceptions or error messages. In the case of hardware failures, when a developer attempts to check the system state, he/she can either check the values of a few input/output devices or use other devices (e.g., a tester or an oscilloscope). Therefore, debugging an embedded system is difficult because it requires the consideration of several factors while estimating the location of errors, and a developer needs to compare both states of hardware and software by his/her eyes. The accuracy of detecting the cause of failure is related to the experience of the developer; this is why novices find it difficult to perform debugging in embedded systems.

From the above reasons, in the embedded system development, the gaze transition of hardware and software is able to be regarded as one of the debugging skills. In general, non-verbal knowledge reflects unconsciously on the eye movement, and

T. Baba (✉) · K. Kiyokawa
Nara Institute of Science and Technology, Ikoma-shi, Nara, Japan
e-mail: baba.takeru.bml@is.naist.jp

E. Makihara · H. Yoneda · K. Ono
Doshisha University, Kyotanabe-shi, Kyoto, Japan
e-mail: emakihar@mail.doshisha.ac.jp

the recognition by the operator is optimized to do tasks. Therefore, eye movement analysis enables us to investigate the techniques of unconscious and non-verbal knowledge of the experts. Hence, the novices are made work efficiency higher by teaching non-verbal knowledge that revealed from eye movement of experts.

This study revealed the differences in the debugging process between novices and experts by analyzing their eye movements during a debugging process. The debugging process of novices shows the difficult points for novices, and the debugging process of experts shows the non-verbal technique in embedded system development. In order to investigate the above debugging characteristics specific to embedded system development, it is important to investigate how a developer compares both hardware and software. The output logs and error messages while debugging, which is generally used for analysis, do not include the data which compares hardware and software. Therefore, we collected the eye movement data for analyzing the non-verbal technique of experts and difficult points of novices.

Additionally, we also investigated which factor the expert developer focuses on while debugging in the early stage to the final stage. We defined the focusing factors by each debugging stage as the modification policy. This paper suggests the expert's modification policy revealed by our investigation as teaching contents for novices. There are several modeling methods for dividing time-series data by unsupervised learning (e.g., GP-HSMM, HDP-HMM+NPYLM). We used Gaussian process-hidden semi-Markov model (GP-HSMM) method that can deal with multiple variables with fast and high accuracy. By clarifying the characteristics of proficiency using the segmentation method, it can reveal the work contents even on a large number of participants. Additionally, analyzing the operation of each segmentation by the Markov model enables us to reveal a modification policy from the eye movement data.

The rest of this paper is structured as follows. Section 2 introduces the features of the eye gaze data and related works on time-series data analysis. Section 3 presents the conditions and results of our experiment. Suggested training contents are introduced in Sect. 4. Finally, the conclusions are presented in Sect. 5.

2 Related Work

2.1 Acquisition and Analysis of Eye Gaze Data

The eye movement of a person changes according to his/her thinking process and skill. It reflects not only his/her intentional behaviors but also the unintentional behaviors (e.g., proficiency and habits). Therefore, eye movement analysis enables us to investigate the techniques of unconscious and non-verbal knowledge of experts.

The time-series data acquired from eye movements indicate various types of information processing. Mostly, we generally recognize an object during a gaze

fixation. We can acquire gaze-fixation data of an operator by using an eye tracker. Thus, we can also analyze his/her recognition activity [1]. By combining gaze-fixation coordinates and data of the object region, the gaze object can be detected. Moreover, by detecting the gaze object, one can analyze its importance, as well as what objects an operator judges the things about.

The features of time-series data can be obtained by modeling rather than using only the original values. Orlov analyzed the proficiency of a developer during source code reading via eye tracking and stated the transition of the functions of the gaze in source code reading by using a hidden Markov model [2]. Busjahn et al. generated a Markov model of the transition of the gaze on functions in source code reading [3]. Hanafusa et al. used a support-vector machine to determine the proficiency level of learners on the basis of an academic record and gaze-distribution data during programming [4].

Related work showed that novices became work efficiency high by teaching the difference between experts and novices eye movement. Ouji et al. analyzed the difference in the review method of source code review by eye movement and investigate whether the efficiency is changed after teaching review method [5]. This paper showed that teaching reading method made participants increase the speed of error detection and detection rate. Matsumoto et al. reported that a support system which shows heat map of fixation by the high-scoring group made the low-scoring groups increase answer speed [6]. In these studies, many of the boundaries between experts and novices are defined on the basis of years of experience and work experience.

2.2 *Time-Series Data and Clustering*

Time-series data represent the data observed with time transition. In time-series data, there exist cases in which the state of each sampling step depends on the previous sampling step. A feature with the aforementioned transition is called a Markov chain. The probability of state transition in the Markov chain is biased, and the data enable us to analyze the relationship of each state.

A common method of investigating the features from the time-series data of eye movement is decided by humans to make subjective judgments about the split points. However, by dividing the time-series data using mathematical method, we can reveal the continuous transition features from large-scale data. The hidden Markov model (HMM) is the most famous segmentation method for time-series data. Since the previous state determines the next state in HMM, it discretizes continuous actions into several states. Therefore, HMM is not able to divide based on a wide range of continuous features because the division points are finely divided. In order to solve this problem, Takahashi et al. divided the time-series data by summarizing the trend of neighboring data that are similar in terms of magnitude, difference, and oscillation [7]. This method is not applicable to handle multidimensional time-series data. Nagasaka et al. used sticky hierarchi-

cal Dirichlet process-HMM and hierarchical Pitman-Yor Language model (HDP-HMM+NPYLM), which is a morphological analyzer based on the n-gram language model for division of operation time-series data [8]. Nakamura et al. proposed a segmentation method named GP-HSMM that combines Gaussian process and hidden semi-Markov model [9]. This paper reported that the segmentation method using GP-HSMM is able to segment motions' continuous state because each clause is represented by the Gaussian process without discretization. This paper also reported that GP-HSMM has higher accuracy than HDP-HMM+NPYLM. However, GP-HSMM could not decide a number of classes automatically and necessary to determine a maximum number of classes. Nagano et al. showed that the number of classes can be extracted automatically by adding hierarchical Dirichlet process (HDP-GP-HSMM) [10]. While the number of classes can be extracted automatically, the execution time is $O(N^3)$, and the computational cost is high. Therefore, it is not suitable for analyzing large-scale data.

Therefore, in this paper, we divided the gaze-object time-series data by using the GP-HSMM method which has a great accuracy and is comparatively faster. We verified that the features of each proficiency are revealed by using GP-HSMM. Subsequently, we looked into the possibility to apply for teaching contents using the revealed features.

3 Experiment

3.1 Objective

According to the literature, eye movement analysis using time-series data modeling can reveal the non-verbal techniques used by experts. Subsequently, a teaching material derived from the aforementioned expert techniques could improve the debugging efficiency of novices. Therefore, we aim to reveal the difference in the debugging process between experts and novices by quantitatively analyzing their eye movements while debugging an embedded system.

To that end, we recorded the eye movements of participants who had experience in embedded system development by using an eye tracker. We investigated the factors for the difference in the debugging process between experts and novices by using the recorded time-series data, which were shown using mathematical models. We recorded the eye movement and operation log as the participants performed debugging. Since a developer acquires proficiency with experience, the eye movement difference caused by proficiency shows a featured skill of embedded system debugging. Therefore, we define the differences in the eye movement by proficiency as a skill of the experts.

3.2 Task

In the experiment, the participants were made to solve a debugging problem, which was to build a system in which an LED blinked during a button was pushed, whether or not the LED turned off during the button was not pushed. We recorded their eye movements and the operational contents of the debugging process until all the bugs were removed. In the initial state of the problem, two bugs were identified in each of the source code and the circuit. The source code contained a bug regarding the operating time delay. Moreover, another contained bug is that subjects need understanding about the circuit button state. The circuit contained bugs associated with checking the connection and required knowledge of the features of the electronic component. The correct and initial circuits are showed in Fig. 1.

3.3 Setup

3.3.1 Configuration

The embedded system debugging experiment was conducted in an environment depicted in Fig. 2. The following are the components in environment:

- Monitor: It is used for source code reading, editing, and obtaining character strings such as compiler logs.
- Instruction sheet: It contains the explanation of the correct version of the system and the circuit. Moreover, it contains a function used in the source code, the way to compile the system, among other information.
- Arduino: This micro-controller controls the electric voltage of the circuit.

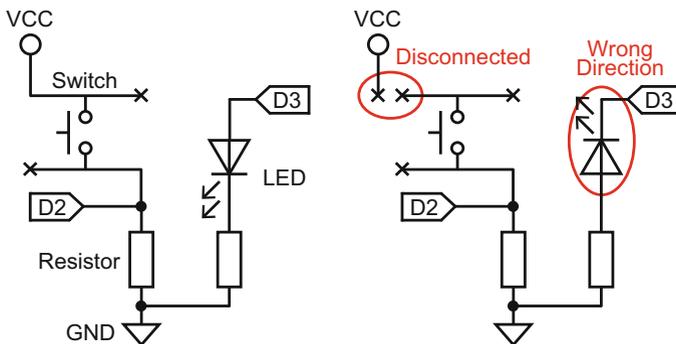


Fig. 1 The correct circuit (left) and the initial circuit with two errors (right)

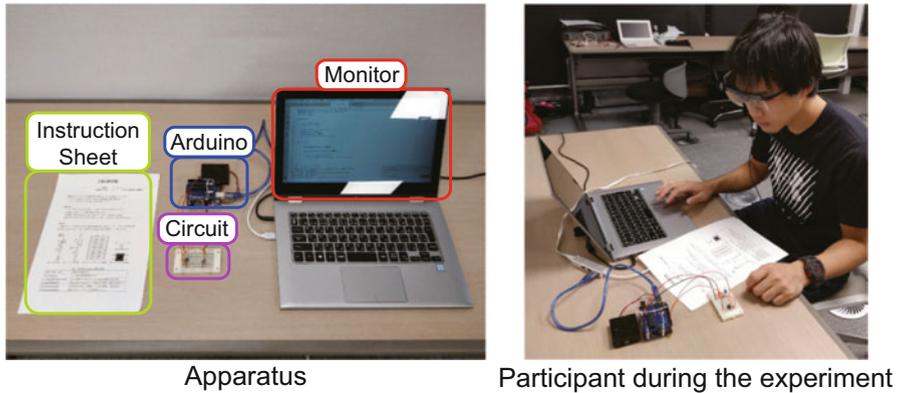


Fig. 2 Apparatus and a participant in the experiment

- **Circuit:** It is related to the digital input/output circuit and has electronic components, such as LED, switch, resistors, and jumper wires.

3.3.2 Recorded Data

We recorded the eye movements at 100 Hz and the user-perspective video of 1920×1080 pixels at 25 fps by using a wearable eye tracker, Tobii Pro Glasses 2. We also recorded timings for gaze fixation which is defined as a slow eye rotation below $30^\circ/s$ for 60 ms [11].

3.4 Procedure

The experiment was held in the following procedure:

- Proc. 1** The participants are provided a system that contained bugs.
- Proc. 2** We explain the goal of the problem to the participants while showing them the correct version of the system.
- Proc. 3** We explain the compiling method and some characteristics in the experiments to the participants. If the participants meet any of the following conditions, we explain the function or components of the circuit by using figures on the instruction sheet:
 - Cond. 1** The participant has never used the Arduino language.
 - Cond. 2** The participant has never implemented a circuit using a circuit diagram but has done so from images.
- Proc. 4** The participant solves the debugging problem.

Proc. 5 The participant fills out a questionnaire regarding the embedded system development (e.g., length of experience, opportunities of develop, and types of micro-controllers used).

3.5 *Participants*

The experiment involved seven undergraduate students who had experience in embedded system development. As discussed in Sect. 2.1, the relevant studies have defined the boundaries between experts and novices based mainly on years of experience. Therefore, experts were defined as those who had considerable experience in embedded system development in this paper. Thus, the participants were divided into experts and novices on the basis of their years of experience. Consequently, the participants were grouped into three experts who experienced over 1 year. Additionally, the experts have used more than four types of micro-controllers and have daily experience of embedded system development, such as those who have taught others to develop embedded systems through club or part-time job. The other four participants were grouped as novices. The novices have used only one type of micro-controller.

3.6 *Results*

3.6.1 *Analysis of the Trend During the Entire Time*

We analyzed the factors causing the difference by proficiency from the eye movement observed while debugging the embedded system. To analyze the time series of the gaze-object data, we built an object detector and automatically performed object detection using the YOLO algorithm [12]. The detector could recognize only four elements discussed in Sect. 3.3.1 (e.g., the monitor and the instruction sheet). Because automatic detection aims to support the analysis results, the detection results were manually verified too. Details, such as the method and precision, are presented in our previous paper [13].

The average problem-solving time and the ratio of gaze upon each object are presented in Table 1. States s_0 through s_3 in Table 1 denote the states when the participants gazed upon the monitor, the instruction sheet, the Arduino, and the circuit, respectively. The experts solved the problem faster than novices. Therefore, we concluded that the proficiency level affected the time required to debug the embedded system.

Next, we investigated the transition of the gaze object at each fixation time while solving the problem. Figure 3 depicts the time series of the gaze-object data for the seven participants. The scale on the horizontal axis represents the number of fixations and that on the vertical axis represents the state of gazing each object. The

Table 1 Average problem-solving time and the ratio of gaze upon each object

	Time [min]	s_0	s_1	s_2	s_3
Experts	13.8	0.627	0.180	0.027	0.166
Novices	32.3	0.589	0.202	0.049	0.160

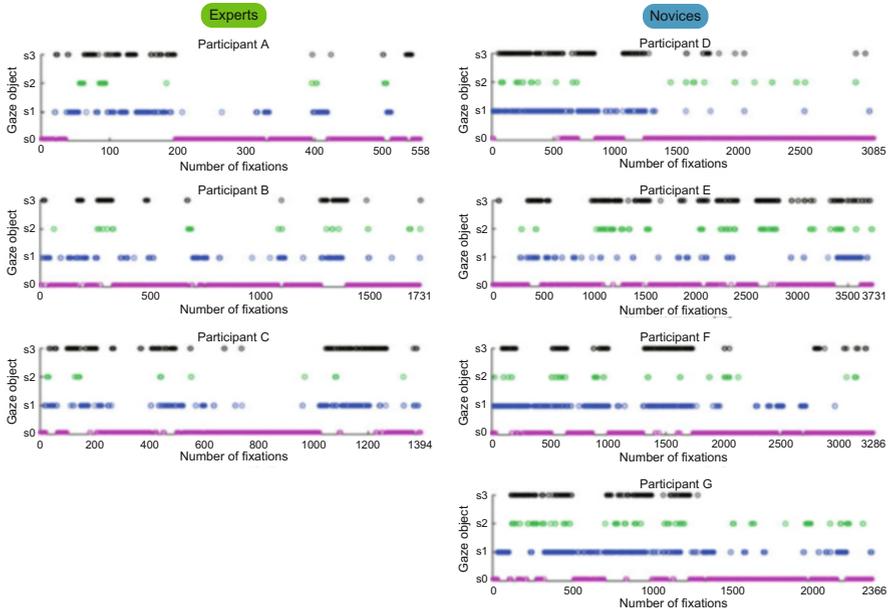


Fig. 3 Time series of the gaze-object data for the seven participants

reason why the number of fixations is the horizontal axis is that we expected that the differences in proficiency occurred from the gaze-object transition, based on one of our previous researches. In our previous paper, the proportion of gaze-object time during the entire experiment did not differ based on proficiency. Another reason is that, in the analysis in the next section, the time of the point process data is not able to apply, but the number of fixation is able. For these reasons, the scale of horizontal axis is unified with the number of fixations. In the figure, Participants A, B, and C represent the eye gaze data of the experts, and Participants D, E, F, and G represent those of the novices.

Previously, one of our studies analyzed the gaze-fixation data for the entire time and the trisected time [14]. In this paper, we use the gaze-fixation data used on the previous paper. From the result of the entire time analysis, no statistically significant difference with Welch’s t-test was observed in the proficiency regarding the gaze ratio and transition. The trend of the entire debugging process revealed that the participants were gazing at the monitor for a long time. Additionally, the gaze transition of the participants mainly occurred via the instruction sheet. According to the result of the trisected time analysis, debugging performed by the experts was divided into the following three stages: circuit debugging in the early stage, source

code debugging in the middle stage, and confirmation of both the circuit and source codes in the final stage. According to the opinions of the experts, the reason for debugging the circuit before debugging the program was as follows: “if a circuit has not been repaired, I cannot debug a program.” Therefore, experts prioritize preparing a state that enables them to see the circuit state. Moreover, according to the opinions of the experts regarding the significance of the confirmation stage, they almost confirmed both the hardware and software simultaneously in the final stage. Hence, experts perform debugging by suspecting the existence of bugs in both the circuit and the source code in the final stage.

However, previous data observation showed that the novices perform debugging in two stages: circuit debugging in the former half and source code debugging in the latter half. Additionally, almost all the novices were found to be confused as they could not formulate the policy and tended to repeat the gaze transition between the circuit and the source code. Therefore, we further analyzed the eye movement division by using a mathematical dividing method.

3.6.2 Partial Analysis Using a Mathematical Division Method

Our previous study showed that by dividing the time series of the eye movement data, the differences are existed in the debugging behaviors by proficient [14]. By grasping the debugging policy of each expert, an efficient debugging process in embedded system development can be revealed. This study investigated the debugging policies of experts and novices using a mathematical division method.

The GP-HSMM method was used to divide the time-series data [9]. The obtained results were different for each execution because the GP-HSMM uses a random number to divide the sections each time. To eliminate randomness, we executed the model five times and then identified the common characteristics during any time range. The time-series data analysis classified by the GP-HSMM was executed in the following procedure:

- Proc. 1** Using the input data of each of the four gaze objects and the data of eye movement distance.
- Proc. 2** Normalizing the data to a real number ranging from 0 to 1.
- Proc. 3** Averaging in blocks and down-sampling time-series data to 100 points.
- Proc. 4** Arranging data into a 100×5 matrix for each participant.
- Proc. 5** Dividing the data into two groups on the basis of the proficiency level of the participants.
- Proc. 6** Performing classification using the GP-HSMM method while considering the following conditions:
 - Cond. 1** $MAXLENGTH = 20$
 - Cond. 2** $MINLENGTH = 5$
 - Cond. 3** $AVERAGELENGTH = 10$
 - Cond. 4** $SKIPLength = 2$

Proc. 7 Replacing and making the class number appropriately via visual inspection because the class number is randomly labeled.

Proc. 8 Modeling using a Markov model and determining the work performed in each class.

Figures 4 and 5 depict the five trial results of division obtained using the GP-HSMM method for experts and novices, respectively. Consequently, the data regarding the eye movements of the experts were divided into three operation classes. The data regarding the eye movements of the novices were divided into four operation classes.

The hidden state of each class must be judged for content by a human. If we focus only on the time proportion for judgment, the bias occurred caused by the time of gaze on “the monitor” takes reading and typing. In contrast, the object of gaze-object transition can represent not only the time of attention but also the relationship between the objects of interest. Hence the transition of gaze object is suitable for judging the content of divided classes. Therefore, we modeled the transition probability using a Markov model for each class. Subsequently, we investigated the type of work that was performed in each class. In this Markov model, the transition probability of an object to the same state is excluded because it is extremely high. In a common Markov model, the transition probability is calculated so that the sum of all the transition probabilities for a state is 1. However, this method calculates the percentage of a large value with a small number of output states (e.g., if a state has only one output, the percentage is 100%). We want the model’s some values to be inconspicuous, since a state which has few outputs is not an important state. Therefore, the transition percentage of our model is calculated such that the sum of all the transition probabilities is 1. Subsequently, we explain the method of

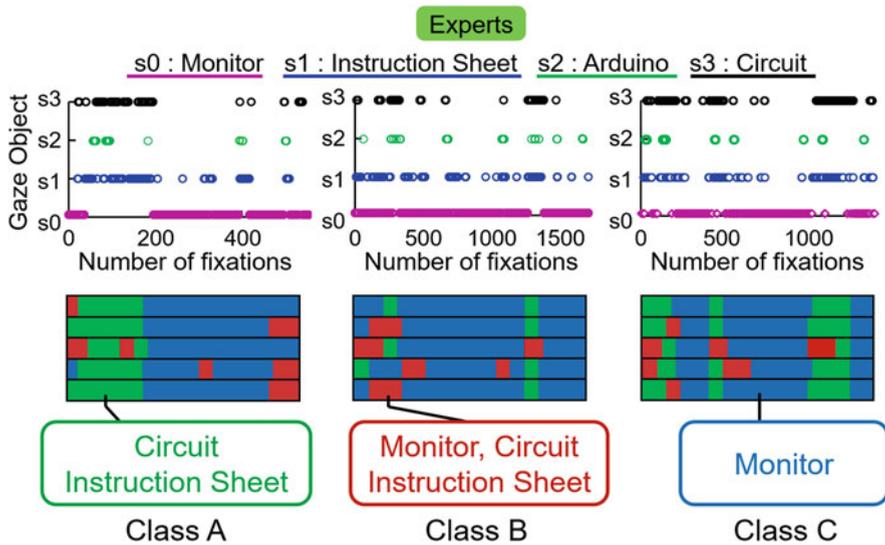


Fig. 4 Results of division via GP-HSMM for experts

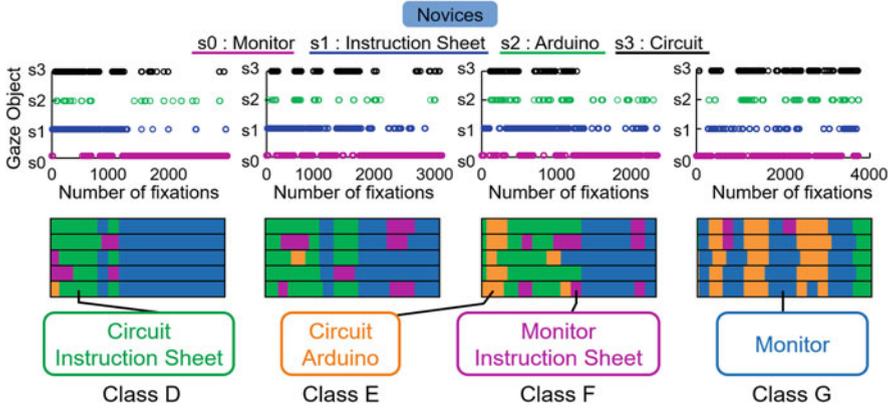


Fig. 5 Results of division via GP-HSMM for novices

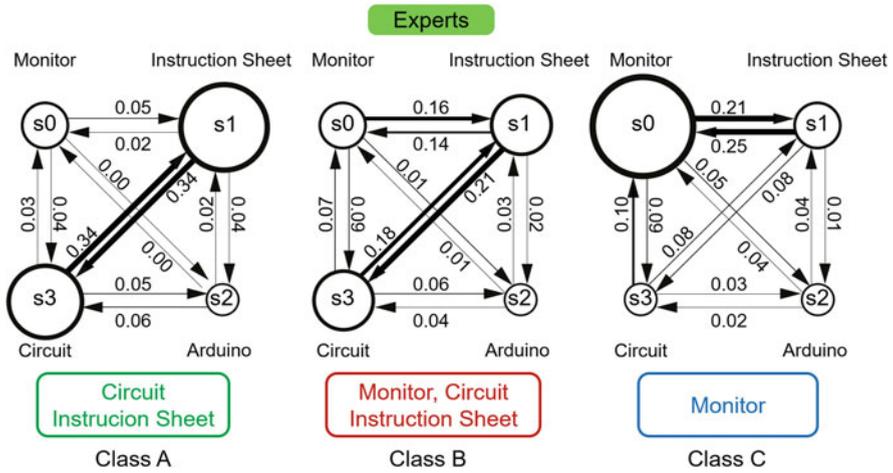


Fig. 6 Markov models of the three expert classes

calculating the transition probability. The term n_{ab} denotes the number of outputs from s_a to s_b ($a, b \in \{0, 1, 2, 3\}$, $a \neq b$). Thus, the number of all the outputs is calculated in Eq. (1):

$$n_{all} = \sum_{x \in X} \sum_{y \in Y} n_{xy} \quad (X, Y = \{0, 1, 2, 3\}, x \neq y) \quad (1)$$

Therefore, the transition probability p_{ab} is shown in below equality:

$$p_{ab} = \sum_{x \in X} \sum_{y \in Y} \frac{n_{ab}}{n_{xy}} = \frac{n_{ab}}{n_{all}} \quad (2)$$

Figures 6 and 7 show the Markov models of the three expert classes and four novice classes. The arrow thickness in these figures represents the transition

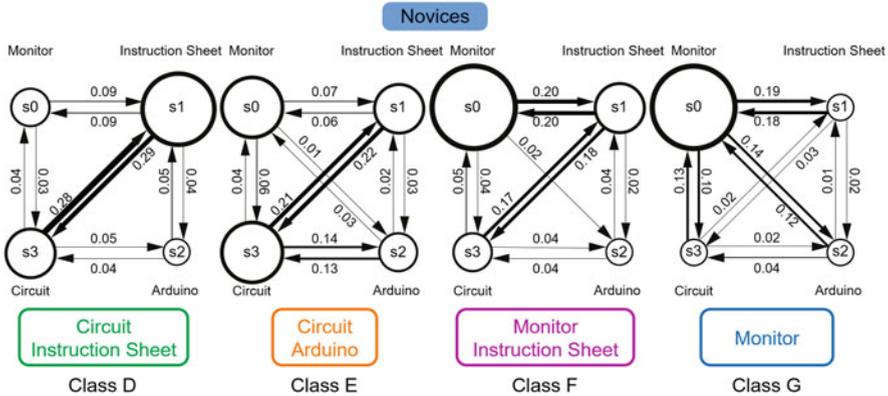


Fig. 7 Markov models of the four novice classes

probability. Thicker arrows signify higher probability. The larger the circle size, the longer the duration of the attention. Among the three expert classes, Class A showed the state of attention on “the circuit and the instruction sheet”; Class B on “the monitor, the circuit, and the instruction sheet”; and Class C on “the monitor.” Among the four novice classes, Classes D and G are equivalent to the expert classes A and C, respectively. Additionally, Class E showed the state of attention on “the circuit and the Arduino” and Class F on “the monitor and the instruction sheet.”

4 Discussion

4.1 Partial Analysis Using the Method of Division

According to the results of the data collected in our experiment, as depicted in Fig. 4 and Fig. 5, the experts performed debugging in three classes and four novices. Regarding the details of the debugging behavior, we estimated the behavior of the experts by focusing on the transition probability from the questionnaire, as depicted in Fig. 6. Consequently, we categorized the behavior of experts into three classes: Class A refers to the behavior of modifying the circuit by referencing the instruction sheet; Class B refers to the behavior of modifying the circuit, as well as the source code, by referencing an instruction sheet; and Class C refers to the behavior of modifying the source code. Considering the aforementioned behaviors of experts, as well as the data presented in Fig. 5, we concluded that experts decide policies in three steps. The experts first check the circuit in its early stage, the source code in the middle stage, and both the circuit and the source code in the final stage for system verification. Additionally, we estimated the behavior of novices by focusing on the transition probability from the instruction sheet depicted in Fig. 7. Consequently,

we categorized the behavior of novices into four classes: Class D refers to the behavior of modifying the source code by referencing the instruction sheet; Class E refers to the behavior of modifying the circuit by comparing Arduino and the circuit; Class F refers to the behavior of modifying the source code by referencing the instruction sheet; and Class G refers to the behavior of modifying the source code. We summarized the classes based on the modification place: Class D and Class E are modification about the circuit; Class F and Class G are modification about the source code. Figure 5 and two summarized modification places show that the novices modify the circuit in the former half and modify the source code in the latter half.

The results for the experts were found to be the same as those in our previous study [14], which divided the debugging behavior into three steps and derived the results. The results therefore confirm that the debugging process of experts consists of three steps.

4.2 Suggesting Teaching Contents for the Education Field

Our investigation revealed two factors that reduce the debugging efficiency of a novice. The first is lack of policy for performing the debugging in advance. In our experiment, the novices repeatedly modified the circuit and the source code in the early stage. Furthermore, they attempted to debug their source code without modifying the output function of the circuit. We conclude that a novice does not decide in advance the method to perform debugging.

If a novice did not have his/her modification policy, he/she may experience some difficulty when operating his/her circuit and in estimating the part of the source code corresponding to the circuit. Therefore, educators must teach novices the following modification policies so that they can also form a modification policy of their own:

- Prioritize preparing a state that enables them to visualize the circuit state.
- First modify the circuit then the source code, and finally check the entire system operation.

The second is lack of sufficient of knowledge regarding the method used for checking the entire system operation. In embedded system development, a developer must check the condition of not only the software but also of the hardware because a single hardware bug can render the entire software operation unclear. Therefore, the educator could provide novices the following method of confirmation used by experts so that the novices can efficiently confirm the condition of the entire system:

- The failure is included in not only the hardware but also the software.

4.3 *Limitation*

In this study, the participants were given a simple debugging assignment that included only a few functions. These functions (e.g., an output function of the circuit and conditions of the source code) are also used in the industry when developing real embedded systems (e.g., electric fans and laser pointers). Therefore, the knowledge and technique of the experts used in our experiment can be considered to be representative of those who are engaged in the development of real embedded systems. However, almost all complex systems are composed of several components (e.g., timers, interruptions, and networks). Moreover, testers and oscilloscopes are the generally used tools for debugging circuits. Consequently, further research is required to reveal the non-verbal techniques of expert developers in more complicated conditions.

5 Conclusion

We focused on a debugging process in embedded system development. We analyzed the differences in the eye movements between novices and experts during a debugging process. In our previous study, in which we quantitatively analyzed the entire time-series data, we did not observe the differences in the debugging skills between the novices and experts [14]. Therefore, in the present study, we attempted to divide the time-series data of the debugging process to estimate the non-verbal techniques used by experts.

Through the present study, we clarified the features of the phases of debugging of both novices and experts. We first divided the time-series eye gaze data collected from the participants and then clustered the data. The results showed that the debugging phases were different between the experts and novices. The experts performed debugging in three phases of circuit debugging, source code debugging and system validation. We obtained similar results from the from the post hoc interview and experiment movies. The reasons are “if a circuit has not been repaired, he/she cannot debug the programs” and that “he/she suspects the existence of bugs in both the circuit and the source code in the final stage.” However, the novices performed debugging in two phases of circuit debugging and source code debugging. Additionally, from the result of the transition of the gaze object and questionnaire, the novices could not decide their debugging policies in the early phase.

Finally, we suggested an advice to novices on the basis of the experimental results to improve their debugging skills. We also proposed a method to distinguish novices who could not decide their modification policies and did not know how to debug their programs. In future work, we will investigate the effects of applying our findings obtained in this study on novices.

Acknowledgments We would like to thank all the educators and students who joined and supported our experiments. Advice and comments given by Professor Miki have been a great help in this research. We also thank Editage (www.editage.com) for English language editing.

References

1. T. Ohno, What can be learned from eye movement? *Under-Understanding Higher Cognitive Processes from Eye Movement Analysis*. Cognitive Studies (2002), pp. 565–579 (in Japanese)
2. P. Orlov, Primary investigation of applying Hidden Markov Models for eye movements in source code reading (2015), pp. 18–20
3. T. Busjahn, C. Schulte, B. Sharif, B. Simon, A. Begel, M. Hansen, R. Bednarik, P. Orlov, P. Ihtantola, G. Alperovich, M. Jetbrains, Eye tracking in computing education, in *Proceedings of the 10th Annual Conference on International Computing Education Research* (2014), pp. 3–10
4. R. Hanafusa, S. Yamagishi, S. Matsumoto, T. Kashima, Automatic classification of eye tracking patterns in reading program based on machine learning, in *Proceedings of the Annual Conference on the Japanese Society for Artificial Intelligence* (2015), pp. 1–2 (in Japanese)
5. S. Oujii, H. Uwanoo, Changes in review efficiency by teaching how to read during code review, in *Research Report Software Engineering (SE)* (2014), pp. 1–8 (in Japanese)
6. K. Matsumoto, T. Wakahara, A Proposal of a programming education support system based on analysis of eye gaze information, in *Proceedings of 80th National Convention of IPSJ* (2018), pp. 721–722 (in Japanese)
7. K. Takahashi, M. Umamo, N. Fujimoto, Partition of time series in temporal axis using hierarchical clustering. *J. Jpn. Soc. Fuzzy Theory Intell. Inf.* **31**, 731–738 (2019) (in Japanese)
8. S. Nagasaka, T. Taniguchi, Motion segmentation with hierarchical Pitman-Yor language model, in *Journal of Proceedings of the Annual Conference of JSAI* (2011), pp. 1–4 (in Japanese)
9. T. Nakamura, T. Nagai, D. Mochihashi, I. Kobayashi, H. Asoh, M. Kaneko, Segmenting continuous motions with hidden semi-Markov models and Gaussian processes. *Front. Neurobotics* **11**, 67 (2017).
10. M. Nagano, Sequence pattern extraction by segmenting time series data using GP-HSMM with hierarchical Dirichlet process, in *International Conference on Intelligent Robots and Systems (IROS)* (2018), pp. 4067–4074
11. A. Olsen, *The Tobii I-VT Fixation Filter*. Organized by Tobii Technology AB (2012)
12. J. Redmon, A. Farhadi, YOLOv3: An Incremental Improvement. ArXiv (2018)
13. H. Yoneda, E. Makihara, T. Baba, Y. Maeda, M. Miki, A proposal for object detection system using YOLO v3 in embedded system development. In the Harris Science Review of Doshisha University (2020), pp. 36–40 (in Japanese)
14. T. Baba, E. Makihara, H. Yoneda, Relationship between eye Gazement and skill level in debug of embedded system development, in *Proceedings of Research Report Software Engineering (SE)* (2019) (in Japanese)

Part VIII
Simulation and Modeling

Modern Control Methods of Time-Delay Control Systems



R. Bars, Cs. Bányász, and L. Keviczky

1 Introduction

It is clear for control engineers that handling time delay requires special attention from the early days of the control history. The time delay is an uncancelable, invariant property of the process. The early goals tried to find design procedures that allow the selection of the regulator quasi independently from the delay. An early success story was the Smith predictor or regulator [1].

Consider a continuous time-delay process given by its transfer function

$$P(s) = P_+(s)\bar{P}_-(s) = P_+(s)e^{-sT_d}; \quad P = P_+\bar{P}_- = P_+e^{-sT_d} \quad (1)$$

where T_d is the time delay and P_+ is the stable and $\bar{P}_- = e^{-sT_d}$ is the *Inverse-Unstable-Unrealizable* (IUU) part of the process, respectively. The original Smith predictor is shown in Fig. 1, where r is the reference signal and y is the process output.

It is easy to check that the SMITH predictor is equivalent to the scheme shown in Fig. 2. This figure clearly shows that the regulator C_+ can be designed to the delay free P_+ , independently of the time delay T_d . This scheme explains why the Smith predictor is also called Smith regulator [2–4]. The whole procedure is, of course, not independent of T_d because the predictor scheme contains block depending on the delay.

R. Bars

Budapest University of Technology and Economics, Budapest, Hungary
e-mail: bars@aut.bme.hu

Cs. Bányász · L. Keviczky (✉)

Institute for Computer Science and Control, SZTAKI, Budapest, Hungary
e-mail: banyasz@sztaki.hu; keviczky@sztaki.hu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_59

811

Fig. 1 The block scheme of the Smith predictor

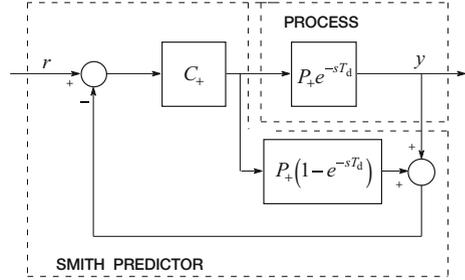


Fig. 2 Equivalent block scheme of the Smith predictor

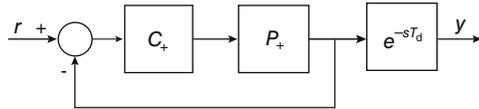


Fig. 3 IMC form of the Smith predictor

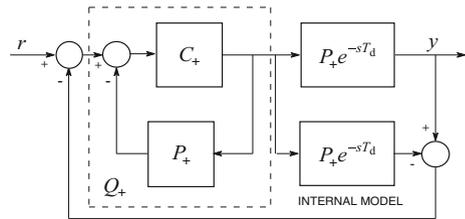
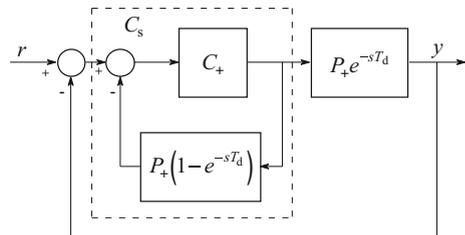


Fig. 4 The resulting closed loop of the Smith predictor



It is possible to redraw the Smith predictor into further schemes, which allow special interpretations. Figure 3. shows another equivalent scheme what corresponds to the well-known *Internal Model Control (IMC)* scheme and principle. Figure 4 presents the resulting closed loop with the serial regulator C_s equivalent to the application of the Smith predictor.

2 The Youla Parameterization

A Youla-parameterized (YP) closed loop [2, 5] is shown in Fig. 5, where e is the error, u is the regulator output, and w is the output disturbance signal.

Here, the plant P is stable and the *All-Realizable-Stabilizing (ARS)* regulator is

Fig. 5 Youla-parameterized closed loop

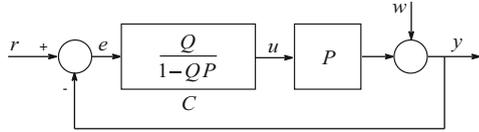
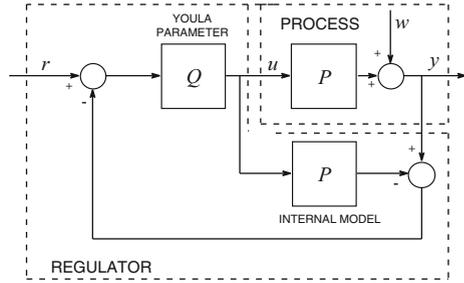


Fig. 6 IMC form of the YP closed loop



$$C = \frac{Q}{1 - QP} \tag{2}$$

The closed-loop transfer function or Complementary Sensitivity Function (CSF)

$$T = \frac{CP}{1 + CP} = QP \tag{3}$$

which is linear in the stable Youla parameter Q .

It is well known that the YP regulator corresponds to the classical IMC structure shown in Fig. 6, where r is the reference signal, u is the regulator output, y is the output signal, and w is the output disturbance signal. If there is no disturbance and the internal model is equal to the process transfer function, the signal fed back to the reference signal is zero, and the forward path QP determines the reference signal tracking. The feedback loop rejects the effect of the disturbance and of the plant/model mismatch.

It can also be well seen that Q_+ in Fig. 3 corresponds to the Youla parameter. For a more detailed comparison, consider the extension of YP regulator for more general case next.

3 A G2DOF Controller for Stable Linear Plants

The first systematic method introducing the *generic two degrees of freedom* (G2DOF) scheme was presented in [2–4, 6]. Two degrees of freedom, 2DOF, implies that the dynamics of reference signal tracking and that of disturbance rejecting are different. This framework and topology are based on the YP providing

ARS regulators for open-loop stable plants and capable of handling the plant time delay, too.

A G2DOF control system for a stable process is shown in Fig. 7.

$$P = P_+ \bar{P}_- = P_+ P_- e^{-sT_d} \tag{4}$$

which is more general than what was used in (1), because here P_+ is stable and *Inverse-Stable-Realizable* (ISR) and P_- is *Inverse-Unstable-Unrealizable* (IUU).

The optimal ARS regulator of the G2DOF scheme can be given by an explicit form

$$C_{opt} = \frac{R_w K_w}{1 - R_w K_w P} = \frac{Q_o}{1 - Q_o P} = \frac{R_w G_w P_+^{-1}}{1 - R_w G_w P_- e^{-sT_d}} \tag{5}$$

where

$$Q_o = Q_w = R_w K_w = R_w G_w P_+^{-1} \tag{6}$$

is the associated optimal *Y parameter*. Furthermore,

$$Q_r = R_r K_r = R_r G_r P_+^{-1}; \quad K_w = G_w P_+^{-1}; \quad K_r = G_r P_+^{-1} \tag{7}$$

The YP regulator (5) can be considered the generalization of the Truxal-Guillemain [2-4, 7] method for stable processes.

It is interesting to see how the transfer characteristics of the closed loop look like:

$$y = R_r K_r P y_r + (1 - R_w K_w P) w = R_r G_r P_- e^{-sT_d} y_r + (1 - R_w G_w P_- e^{-sT_d}) w = y_r + y_d \tag{8}$$

where y_t is the tracking (servo) and y_d is the regulating (or disturbance rejection) independent behaviors of the closed-loop response, respectively. So the delay e^{-sT_d} and P_- cannot be eliminated, and consequently the ideal design goals R_r and R_w are biased by $G_r P_-$ and $G_w P_-$. Here, R_r and R_w are assumed stable and usually strictly proper transfer functions that are partly capable of placing desired poles in the tracking and the regulatory transfer functions; furthermore, they are usually referred to as reference signal and output disturbance predictors. They can even be

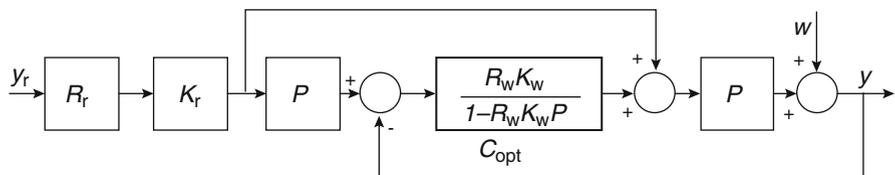


Fig. 7 The generic 2DOF (G2DOF) control system

called as reference models, so reasonably $R_r(\omega = 0) = 1$ and $R_w(\omega = 0) = 1$ are selected. The unity gain of R_w ensures integral action in the regulator, which is maintained if the applied optimization provides $G_w P_-(\omega = 0) = 1$.

The role of R_r and R_w (predictors or filters) is threefold. They prescribe the tracking and regulatory properties of the control loop. They influence the magnitude of the actuating signal and also influence the robustness properties of the control system.

An interesting result was found [8] that the optimization of the G2DOF scheme can be performed in H_2 and H_∞ norm spaces by the proper selection of the serial embedded filters G_r and G_w , attenuating the influence of the invariant process factor P_- . Using H_2 norm, a Diophantine equation (DE) should be solved to optimize these filters. If the optimality requires a H_∞ norm, then the Nevanlinna-Pick (NP) approximation is applied.

After some straightforward block manipulations, the G2DOF control system can be transformed to another form shown in Fig. 8, which is the generalized version of the classical IMC scheme in Fig. 6.

4 Smith Predictor as a Subclass of G2DOF Controllers

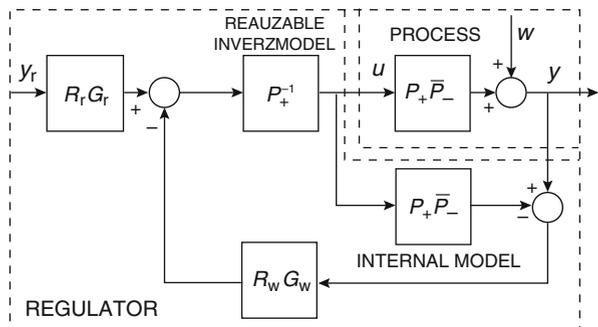
The previous two sections clearly show that the Smith predictor is a special subclass of the G2DOF controllers with a YP parameterized regulator

$$Q_+ = \frac{C_+}{1 + C_+ P_+} = \frac{C_+ P_+}{1 + C_+ P_+} P_+^{-1} = \frac{L_+}{1 + L_+} P_+^{-1} = R_+ P_+^{-1} \tag{9}$$

if C_+ is stabilizing P_+ , i.e., the delay free part of the process. Here, the special CSF

$$T_+ = R_+ = \frac{L_+}{1 + L_+} \tag{10}$$

Fig. 8 The generalized IMC form of the G2DOF control system



characterizing the closed loop in Fig. 2 is the reference model R_+ and $L_+ = C_+P_+$ is its loop transfer function.

It is also easy to see that the resulting serial regulator of the Smith predictor in Fig. 4 is

$$C_s = \frac{Q_+}{1 - Q_+P_+e^{-sT_d}} = \frac{C_+}{1 + C_+P_+(1 - e^{-sT_d})} = C_+K_S \quad (11)$$

This formula presents the possible way of realization for a continuous-time (CT) case. Here K_S denotes a serial factor modifying the original C_+ regulator of the SMITH predictor

$$K_S = \frac{1}{1 + C_+P_+(1 - e^{-sT_d})} = \frac{1}{1 + L_+(1 - e^{-sT_d})} \quad (12)$$

At the stability limit cross over frequency ω_c , where $L_+ = -1$ the factor K_S takes a considerable positive phase advance into the closed-loop

$$K_S = \frac{1}{1 + (-1)(1 - e^{-sT_d})} = \frac{1}{1 - 1 + e^{-sT_d}} = e^{sT_d} \left(\omega_c = e^{j\omega_c T_d} \right) \quad (13)$$

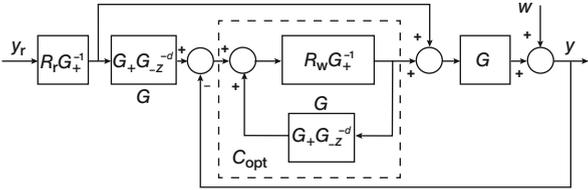
This is the simple physical explanation of the success of the Smith predictor [9].

Some early evaluations state that unfortunately the Smith predictor is only good for tracking and not for disturbance rejection. This evaluation is wrong. The Smith regulator was proposed for a one degree of freedom (1DOF) closed loop, so it is naturally not for 2DOF purposes. The real problem of the Smith regulator is that it allows the design of the closed loop only via an indirect way by selecting $R_+ = T_+$, while the design procedure of the G2DOF scheme gives a direct procedure to design the independent tracking and disturbance rejection properties. This means that the original idea of Smith was that a classical design of T_+ is necessary for proper application. One must know that the Youla parameterization and its application for regulator design were unknown to Otto Smith when he invented his predictor.

5 The Discrete-Time Version of G2DOF Controllers

Although (11) suggests a proper way of how to realize the Smith regulator, it is not realistic to build any regulator containing the e^{-sT_d} delay element for continuous time case. In practice, only the discrete-time (DT) version can be applied by computer realization. Consider the DT model of the CT process in the form of its pulse transfer function given by

Fig. 9 Discrete-time G2DOF control system for the suboptimal $G_r = G_w = 1$ case



$$\begin{aligned}
 G(z^{-1}) &= G_+(z^{-1}) \overline{G}_-(z^{-1}) = G_+(z^{-1}) G_-(z^{-1}) z^{-d} \\
 G &= G_+ \overline{G}_- = G_+ G_- z^{-d}
 \end{aligned}
 \tag{14}$$

where G_+ is stable and ISR, G_- is IUU, and z^{-d} corresponds to the discrete time delay, where d is the integer multiple of the sampling time. (In a practical case, the factor G_- can incorporate the underdamped zeros and the neglected poles providing realizability, too). The optimal ARS regulator of the G2DOF scheme can be given now by

$$C_o = \frac{R_w K_w}{1 - R_w K_w S} = \frac{Q_o}{1 - Q_o G} = \frac{R_w G_w G_+^{-1}}{1 - R_w G_w G_- z^{-d}}
 \tag{15}$$

which corresponds to the CT case of (5); furthermore (6) and (7) are formally exactly the same for DT case. The transfer characteristic of the closed loop is now

$$\begin{aligned}
 y &= R_r K_r G y_r + (1 - R_w K_w G) w = R_r G_r G_- z^{-d} y_r \\
 &\quad + (1 - R_w G_w G - z^{-d}) w = y_t + y_d
 \end{aligned}
 \tag{16}$$

Because the optimization of the embedded filters G_r and G_w requires special knowledge and practice of getting the solution from a DE and NP approximation, suboptimal design is mostly applied assuming $G_r = G_w = 1$. In such cases, the influence of the invariant process factors is not attenuated at all, so they appear in the closed-loop characteristics (15) directly. Such G2DOF control scheme is shown in Fig. 9.

It follows from the above discussion that it is not necessary to apply the classical Smith predictor principle, instead it is more effective to use the regulator design procedure of the G2DOF controller scheme.

6 Simple Examples

6.1 Example 1

Consider a very simple first-order CT time-delay process

$$P = \frac{1}{1 + 10s}e^{-5s}; \quad P_+ = \frac{1}{1 + 10s}; \quad \bar{P}_- = e^{-5s}; \quad P_- = 1 \quad (17)$$

The tracking and disturbance rejection reference models are

$$R_r = \frac{1}{1 + 4s} \text{ and } R_w = \frac{1}{1 + 2s} \quad (18)$$

Here $P_- = 1$; therefore, $G_r = G_w = 1$ is the optimal selection for the embedded filters.

Design a Youla-parameterized optimal regulator.

$$\begin{aligned} C_{\text{opt}} &= \frac{R_w G_w P_+^{-1}}{1 - R_w G_w P_- e^{-sT_d}} = \frac{1}{1 - R_w e^{-sT_d}} R_w P_+^{-1} \\ &= \frac{1}{1 - \frac{1}{1+2s} e^{-5s}} \frac{1+10s}{1+2s} = \frac{(1-2s)(1+10s)}{1+2s - e^{-5s}} \end{aligned} \quad (19)$$

and the optimal serial compensator is

$$R_r K_r = R_r G_r P_+^{-1} = R_r P_+^{-1} = \frac{1 + 10s}{1 + 4s} \quad (20)$$

Both transfer functions are realizable. Because $C_{\text{opt}}(s = 0) = \infty$, the regulator is integrating obtained from the condition $R_w(s = 0) = 1$. The optimal final closed loop is shown in Fig. 10. Although all blocks are realizable in this scheme, it is very unrealistic that the real CT models of the true process are applied in a practical application. Here the real difficulty is the realization of the time delay. So this example stands only to represent the YP-based G2DOF design procedure.

It is easy to check that the closed-loop characteristics is

$$\begin{aligned} y_{\text{opt}} &= R_r e^{-sT_d} y_r + (1 - R_w e^{-sT_d}) w \\ &= \frac{1}{1+4s} e^{-5s} y_r + \left(1 - \frac{1}{1+2s} e^{-5s}\right) w \end{aligned} \quad (21)$$

according to the general theory.

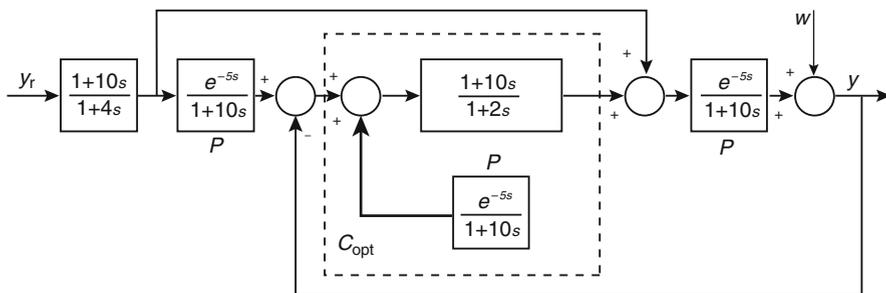


Fig. 10 The designed optimal closed loop of the example

6.2 Example 2

Consider now the DT model of a very simple first-order time-delay process

$$G = \frac{0.2z^{-1}}{1 - 0.8z^{-1}}z^{-3} = \frac{0.2z^{-4}}{1 - 0.8z^{-1}}; \quad G_+ = \frac{0.2z^{-1}}{1 - 0.8z^{-1}} \quad \text{and} \quad G_- = 1 \tag{22}$$

It is required to speed up the process by a closed loop. Design a YP controller. Select the reference models

$$R_r = \frac{0.8z^{-1}}{1 - 0.2z^{-1}} \quad \text{and} \quad R_w = \frac{0.5z^{-1}}{1 - 0.5z^{-1}} \tag{23}$$

Because $G_- = 1$, there is no optimization task, so the selections $G_r = 1$ and $G_w = 1$ are optimal. The optimal regulator is

$$\begin{aligned} C_{\text{opt}} &= \frac{R_w G_w G_+^{-1}}{1 - R_w G_w G_- z^{-d}} = \frac{1}{1 - R_w z^{-d}} R_w G_+^{-1} \\ &= \frac{1}{1 - \frac{0.5z^{-1}}{1 - 0.5z^{-1}} z^{-3}} \frac{0.5z^{-1}}{1 - 0.5z^{-1}} \frac{1 - 0.8z^{-1}}{0.2z^{-1}} = \frac{2.5(1 - 0.8z^{-1})}{1 - 0.5z^{-1} - 0.5z^{-4}} \end{aligned} \tag{24}$$

and the serial compensator is

$$R_r G_+^{-1} = \frac{0.8z^{-1}}{1 - 0.2z^{-1}} \frac{1 - 0.8z^{-1}}{0.2z^{-1}} = \frac{4(1 - 0.8z^{-1})}{1 - 0.2z^{-1}} \tag{25}$$

The optimal final closed loop is shown in Fig. 11. Observe that $C_{\text{opt}}(z = 1) = \infty$, i.e. the regulator is an integrating one, which follows from the condition $R_w(z = 1) = 1$. The closed-loop characteristic is

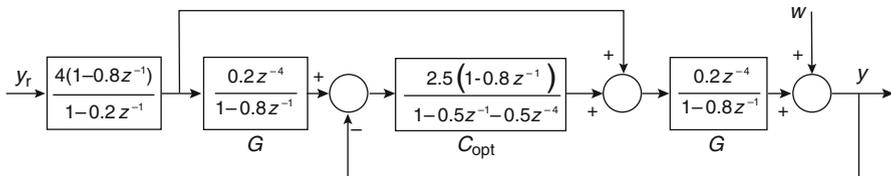


Fig. 11 The designed optimal closed loop of the example

$$\begin{aligned}
 y_{\text{opt}} &= R_r z^{-d} y_r + (1 - R_w z^{-d}) w = \frac{0.8z^{-1}}{1-0.2z^{-1}} z^{-3} y_r + \left(1 - \frac{0.5z^{-1}}{1.0.5z^{-1}} z^{-3}\right) w \\
 &= \frac{0.8z^{-4}}{1-0.2z^{-1}} y_r + \left(1 - \frac{0.5z^{-4}}{1-0.2z^{-1}}\right) w
 \end{aligned} \tag{26}$$

which exactly corresponds to our design goals.

This example shows that there is no applicability problem for DT regulator design. These filters are easy to be realized in a computer controlled system.

6.3 Example 3

The continuous first-order plant with significant time delay is given by the transfer function

$$P(s) = \frac{1}{1 + 10s} e^{-30s} \tag{27}$$

The plant is sampled with sampling time $T_s = 5$ sec, and a zero-order hold is applied at its input. Let us design a PI controller ensuring about 60° of phase margin, a Smith predictor, and a Youla-parameterized controller. Compare the reference signal tracking and output disturbance rejection behavior of the three control systems. Demonstrate the effect of time-delay mismatch.

The pulse transfer function of the plant is

$$G(z) = \frac{0.3935}{z - 0.6065} z^{-6} \tag{28}$$

The pulse transfer function of the PI controller [10] applying pole cancellation with a gain ensuring the required phase margin is

$$C_{\text{PI}}(z) = 0.204 \frac{z - 0.6065}{z - 1} \tag{29}$$

The Smith predictor controller C_+ is designed for the delay-free process as a PI controller and it is obtained as

$$C_+(z) = 2.5 \frac{z - 0.6065}{z - 1} \tag{30}$$

Then, it is transformed to the Smith predictor form according to the discretized version of (11).

$$C_s(z) = \frac{2.5z^7 - 1.516z^6}{z^7 - 0.01636z^6 - 0.9837} \tag{31}$$

In the case of the Youla- parameterized controller, let us choose the disturbance filter

$$R_w(s) = \frac{1}{1 + 5s} \tag{32}$$

and the reference filter as

$$R_r(s) = \frac{1}{1 + 8s} \tag{33}$$

whose pulse transfer functions are

$$R_w(z) = \frac{0.6321}{z - 0.3679} \quad \text{and} \quad R_r(z) = \frac{0.4647}{z - 0.5353}, \tag{34}$$

respectively.

The Youla parameter supposing $G_r = G_w = 1$ is $Q(z) = R_w(z)G_+^{-1}(z) = \frac{0.6321}{z-0.3679} \cdot \frac{z-0.6065}{0.3935}$ (35)

Figure 12 shows the step response and a shifted step disturbance rejection of the three controllers.

It is seen that in case of significant time delay, Smith predictor and Youla-parameterized controllers ensure significant acceleration compared to the PI controller.

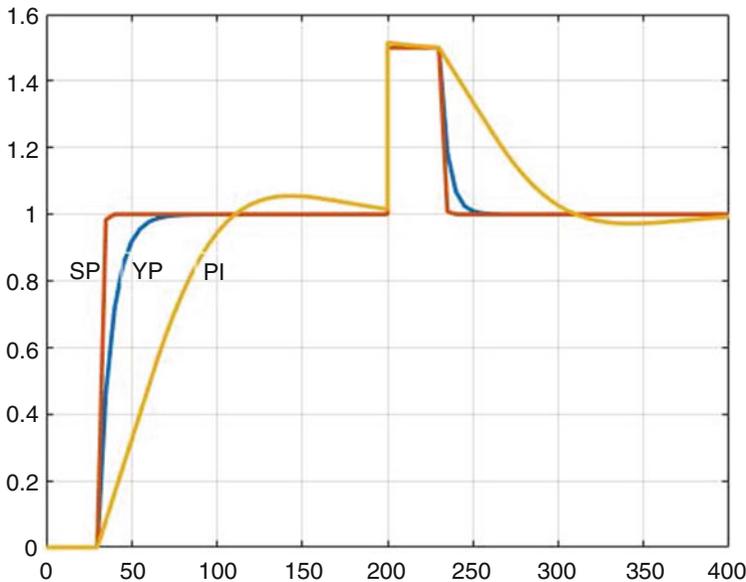


Fig. 12 Step response and disturbance rejection dynamics of the PI, Smith, and Youla controllers

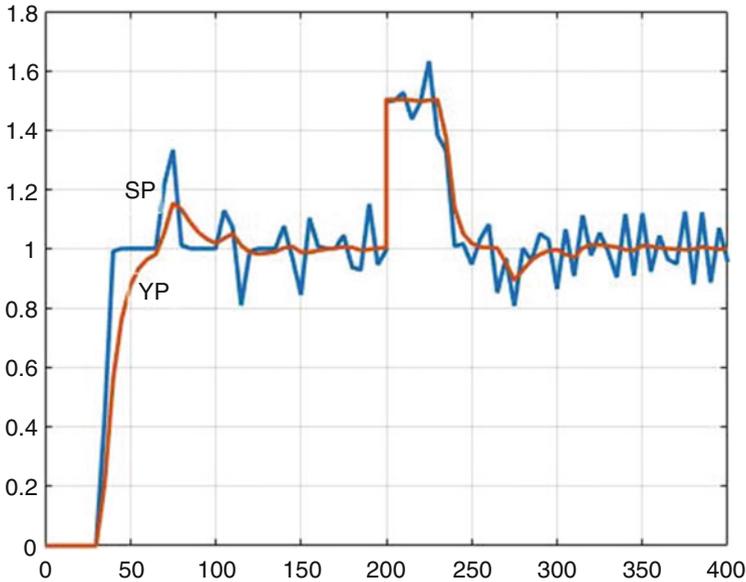


Fig. 13 The effect of time-delay mismatch in case of the Smith and Youla controllers

Figure 13 demonstrates the effect of time-delay mismatch in the case of Smith and the Youla controllers. The time delay of the model is 30, while the time delay of the process is 33.

It is seen that the Youla-parameterized controller tolerates much better the inaccuracy of the parameter than the Smith predictor. While the Smith predictor is very sensitive to the inaccuracies in the parameters (it is not robust), the filters in the Youla-parameterized controller can be designed for robust behavior [11].

These are, of course, very simple examples standing only to present the simplicity of the G2DOF controller scheme, which should replace the classical approach of a Smith predictor.

7 Conclusions

The Smith predictor is a classical method of handling time delay in closed-loop control design. It is shown that this method is a subclass of the YP-based G2DOF control scheme. An obvious drawback of the Smith predictor is that the closed-loop properties can-not be designed directly using simple algebraic methods, which is possible in the G2DOF structure. The G2DOF scheme allows even the optimal attenuation of the invariant process factors. The appropriate choice and design of the filters allow to influence such important properties as performance and robustness.

So this chapter suggests to use the newer methodology to design DT controllers for time-delay processes.

The role of the Smith predictor remains important in the history of control engineering because it was one of the first, easy-to-use, and widely applied method to simply eliminate the influence of the delay in the design of closed-loop control properties. Nevertheless, this method is sensitive to the accurate knowledge of the time delay.

The recent theoretical developments and easily applicable algebraic design methods allow to use more effective and more general controller design procedures.

References

1. O.J.M. Smith, Closed control of loops with dead time. *Chem. Eng. Progress*, **53**, 217 (1957)
2. L. Keviczky, C. Bányász, *Two-Degree-of-Freedom Control Systems (The Youla Parameterization Approach)* (Elsevier, Academic Press, 2015)
3. L. Keviczky, R. Bars, J. Hetthéssy, C. Bányász, *Control Engineering* (Springer, 2018)
4. L. Keviczky, R. Bars, J. Hetthéssy, C. Bányász, *Control Engineering: MATLAB Exercises* (Springer, 2018)
5. J.M. Maciejowski, *Multivariable Feedback Design* (Addison Wesley, 1989)
6. L. Keviczky, Combined identification and control: Another way, (Invited plenary paper.) in *5th IFAC Symposium on Adaptive Control and Signal Processing, ACASP'95*, Budapest, Hungary, 13–30, 1995
7. I.M. Horowitz, *Synthesis of Feedback Systems* (Academic Press, New York, 1963)
8. L. Keviczky, C. Bányász, Optimality of two-degree of freedom controllers in H_2 - and H_∞ -norm space, their robustness and minimal sensitivity, in *14th IFAC World Congress*, F, 331–336, Beijing, PRC, 1999
9. K.J. Åström, B. Wittenmark, *Computer Controlled Systems* (Prentice-Hall, 1984), p. 430
10. N. Tan, Computation of stabilizing PI and PID controllers for process with time delay. *ISA Trans.* **44**, 213–223 (2005)
11. C. Bányász, L. Keviczky, R. Bars, Influence of time delay mismatch for robustness and stability, in *IFAC TDS*, Budapest, Hungary, 248–253, 2018

An Interactive Software to Learn Pathophysiology with 3D Virtual Models



Abel A. Reyes, Youxin Luo, Parashar Dhakal, Julia Rogers, Manisa Baker, and Xiaoli Yang

1 Introduction

Incorporating innovative teaching strategies using technology needs to be implemented within nursing education to ameliorate cognitive learning, enhance patient care, and improve patient outcomes. In graduate nursing education, the development of consistent clinical knowledge and critical reasoning is mandatory to prepare advanced practice nurses for the workforce. Graduate nursing students must learn to identify disease progression, understand the basis of ordering diagnostic testing, and comprehend the rationale of specific treatment plans. To ensure comprehension, the graduate advanced practice nursing students require an in-depth understanding of pathophysiology. The use of a 3D interactive visualization aids in building a strong foundation of understanding homeostasis and recognizing disease states and progression.

Since the National League for Nursing (NLN) made a call for innovation in nursing education [1, 4], the emphasis has been on the development and implementation of innovative pedagogical methods. The use of active learning techniques has been highly recommended by the NLN. Any learning activity where students interact within the learning process is known as active learning

A. A. Reyes (✉) · Y. Luo · X. Yang

Electrical and Computer Engineering, Purdue University Northwest, Hammond, IN, USA
e-mail: areyesan@pnw.edu; luo300@pnw.edu; yangx@pnw.edu

P. Dhakal

Electrical and Computer Science Engineering, The University of Toledo, Toledo, OH, USA
e-mail: Parashar.dhakal@utoledo.edu

J. Rogers · M. Baker

College of Nursing, Purdue University Northwest, Hammond, IN, USA
e-mail: jlrogers@pnw.edu; baker417@pnw.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_60

825

[2]. Students become actively involved in the learning process by participating, analyzing, synthesizing, and evaluating content. Active learning is an investment of energy in all phases of the learning process and is more apt to stimulate higher cognitive processes, such as those associated with critical thinking. Overall, the use of interactive advanced learning tools provides graduate nursing students an edge in comprehending difficult coursework material [3]. Research has shown that technology-based learning improves comprehensive learning, knowledge retention, and long-term memory recall [5–8]. It is imperative to create a sense of gratification for students during the learning experience [9]; thus, a 3D interactive visualization not only can contribute to improve pedagogical strategies but also can ensure cognitivism as the learning theory behind the new teaching methodologies [13, 14]. Research has shown that interactive visualization tutorials are more effective in cognitive learning as compared to a script-based approach. This is due in part because active learning tools support both the cognitive and the affective domains [13, 15, 16]. This work discusses the implementation of an interactive 3D visualization tool for a graduate nursing pathophysiology course. The paper is structured as follows: background and motivations, where a summary of related works is provided along with the motivations behind this work; application design, where the technical details regarding the work is explained along with the description of the tools used in its development; next a brief coverage of the nursing topics are delivered along with the reasons behind the selection of those topics; finally, conclusions are provided along with suggested ideas for future work.

2 Background and Motivations

This work was completed through a collaboration among researchers from the department of computer engineering and the college of nursing. The goal is to implement an interactive 3D visualization software tool to improve graduate nursing students understanding and retention of knowledge regarding the pathophysiology of disease processes. Pathophysiology is the study of processes associated with disease or injury and is the basis of advanced nursing practice. Prior to the development of the 3D interactive visualization software tool for nursing education, several related works were reviewed. The extensive and detailed review of literature provided a comprehensive understanding of how to develop an effective active learning visualization tool for graduate nursing students. A brief description of the literature review is presented in this section. Padilha [9] worked on the evaluation of a virtual clinical simulation within the classroom and determined simulation techniques aid in knowledge retention and clinical reasoning and provided, at minimum, satisfactory learning experiences for the students. Similarly, Kluger [10] in his work concluded that new technologies, including immersive virtual reality (VR) simulators and interactive software tools, have the potential to improve healthcare educational programs to produce better prepared health professionals for the real world. The researchers further reported, from a survey taken in 2017, that

about 65% of nursing educational programs use simulations within the classroom. However, most simulations currently in use within nursing education are with the use of mannequins and hospital-based patient case scenarios, not 3D visualization simulations for understanding the pathophysiology of diseases. Choi [11] discussed the satisfactory use of a VR simulator for a single process in the healthcare training, with the goal to provide a self-learning tool for students, without additional material expenses. In addition, Harjanto [12] claimed the e-learning method as the new standard in the education for nurses, since this offers flexibility in the learning process and has had good results. New technologies, such as computer simulators and VR trainers, are an innovative means for complimentary pedagogy instruction in graduate nursing education. To the best of our knowledge, there is null evidence of an interactive 3D visualization project implemented to cover the complete curricula in a specific graduate nursing course. Overall, there is a lack of research on active learning tools used in the improvement of advanced practice nursing students comprehension of disease processes and its effect on human body's natural homeostatic state. Furthermore, there is null evidence on a 3D visualization that addresses the differentiation of disease processes. There remains a gap in the research on 3D visualization tutorials that explain and display the actual pathophysiologic processes that occur with specific treatments for disease related to specific patient populations. Based on the aforementioned review of literature, the implementation of an interactive 3D visualization software tool for graduate nursing education is proposed. The pedagogical tool is expected to stimulate the cognitive learning of the students, create a satisfactory experience, report long-term knowledge retention, and perform as an active learning technique along the education process. The project is expected to be implemented, tested, and deployed for the population of graduate nursing students at Purdue University Northwest (PNW), focusing in advanced pathophysiology and serving as a complement to the traditional teaching methodology. This interactive visualization will be utilized in the face-to-face environment as well as the online environment.

Algorithm 1 Project workflow

```
1: while All requirements are not met do  
2:   if Deliverable version is complete then  
3:     Test with the target users  
4:     Collect feedbacks  
5:   else  
6:     for Every Scene in Unity do  
7:       for Every 3D Model do  
8:         while 3D model != 3D model expected do
```

```
9:         Texturing the 3D model
10:        Lightning the 3D model
11:        Render the 3D model
12:        if Animation is required then
13:            Bake the animation
14:        end if
15:        end while
16:        Send 3D model to Unity as an asset
17:    end for
18:    Integrate multimedia content
19:    Integrate Scripts
20:    for Every asset in the scene do
21:        if asset require component or script then
22:            Add component or script
23:        end if
24:    end for
25: end for
26: end if
27: end while
```

3 Application Design

In our work, a novel interactive tool based on visualization has been proposed, which, in its framework, includes most of the characteristics retrieved from the research performed. In particular, the academic material, covered in the application, will be focused in the delivery of pathophysiology information, for different diseases among the several systems in the human body. During the development, we ensured that the tool helps the user understand information which allows for greater retention of material. The interactive 3D visualization was designed to assist the learner in development of reasoning skills and comprehension and relate to practical real-world healthcare provider experiences.

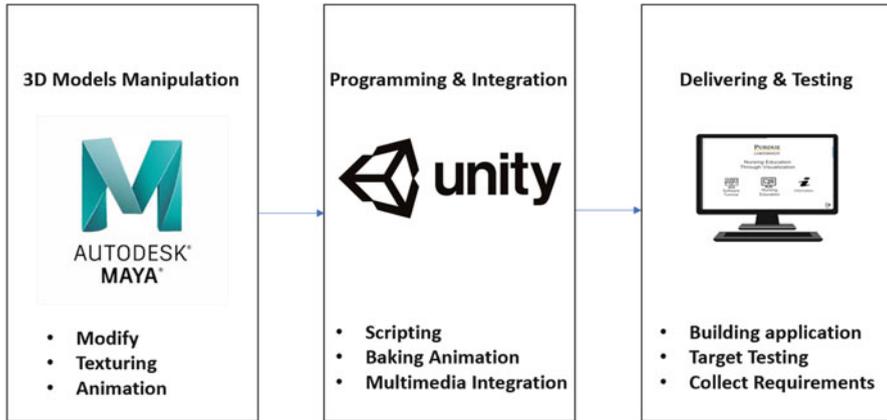


Fig. 1 Software workflow

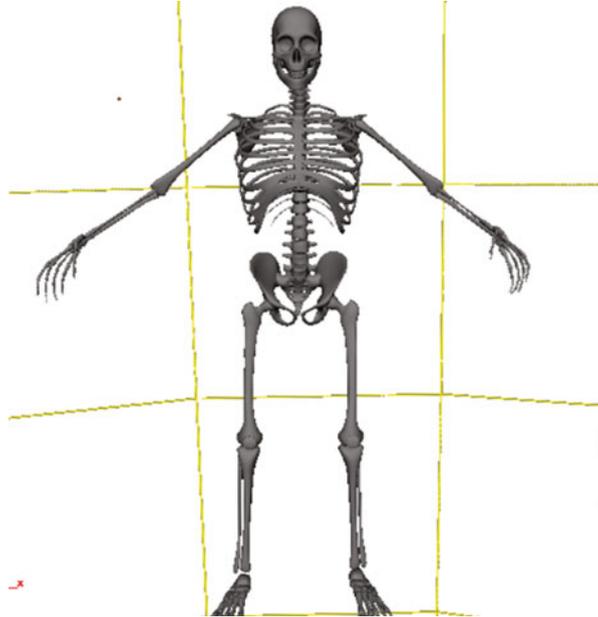
To develop the software, a workflow was proposed and used as guide for the project. Figure 1 shows the workflow used for this project. It consists of three major sections: 3D model(s) manipulation, programming and integration, and delivering and testing. The process followed in this workflow has been detailed in Algorithm 1.

3.1 3D Model Manipulation

In the first section of the workflow, the manipulation of the 3D models displayed in the application was performed. There were two kinds of 3D models: CAD solid model and polygon model. The characteristics between both the model formats differ as follows: a solid model shows a better accuracy in details while compromising flexibility to work with different tools. On the other hand, polygonal models have a more suitable format, allowing it to be easily exported to the IDE used in the development of this application, without compromising the quality of the models. Polygonal models fit our needs in the project, since they are widely compatible with different development tools. Further, they also showed satisfactory quality in displaying minute details by delivering realistic and accurate models.

For the manipulation of the 3D models, a particular 3D graphic computer software named Autodesk Maya [17] was considered because of the availability of large amounts of features within the software and because of its user-friendly environment. This allowed the learning curve to use the tool to be almost imperceptible. Maya allowed us to manipulate the model, customize the meshes, create animations, and export them to the software engine to develop the interactive application. In Fig. 2, an example of the polygonal models used in the development of the interactive application within the environment of Maya has been depicted.

Fig. 2 Polygon 3D model in Autodesk Maya



In this stage, the 3D models were manipulated, customized, and suitably used in the following stages. During this process, accuracy of the models was considered of the utmost importance. The visualization created the most realistic version of the 3D model, using texture, lightning, and rendering features available in the graphic software. Part of production for the animations was done using this software; however, most of them were developed in the next stage: programming and integration.

3.2 *Programming and Integration*

The second stage is basically the software development of the interactive application. To describe this section, a software development life cycle (SDLC) model was used as a framework which gives a better idea of the operation performed on each phase within this stage, especially for documentation and maintenance purpose. The most suitable model was the Agile model [18]. Since this is an interdisciplinary project (involving computer engineering and nursing), the Agile models allowed us to handle, in a better way, the change in the requirements, which is very common in educational-based projects.

Figure 3 shows the phases followed using the Agile model. It is important to notice that requirements are constantly feeding the model and after review any

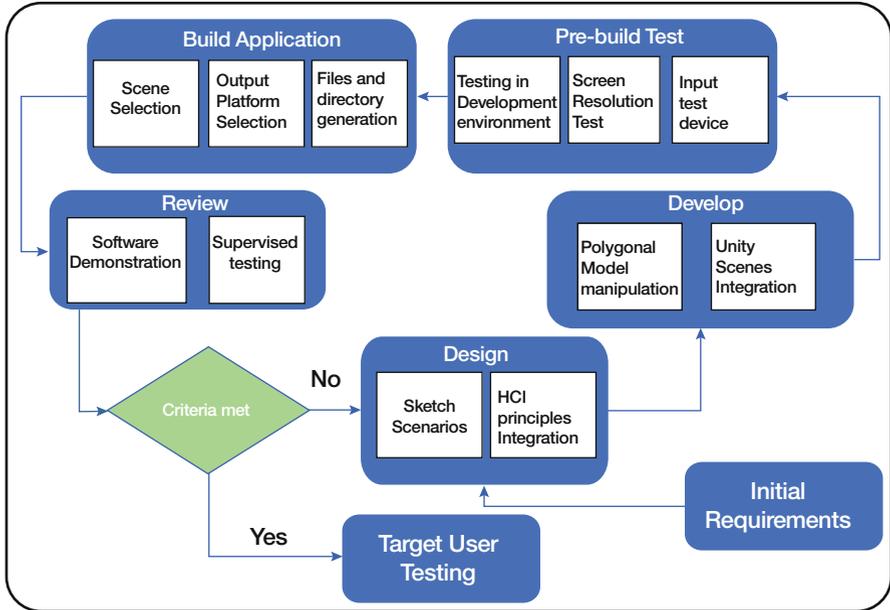


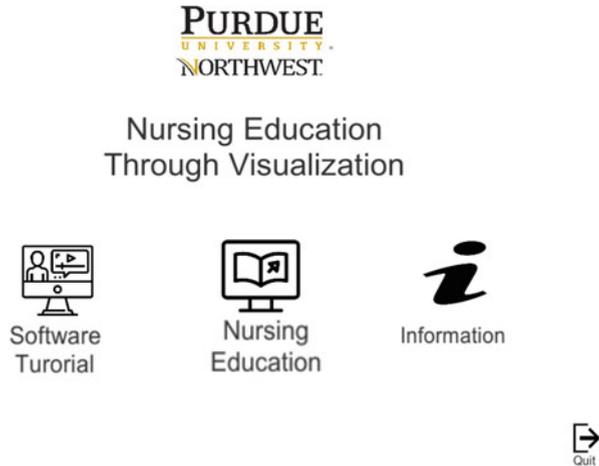
Fig. 3 SDLC framework – Agile model used in the software development stage

module implemented from the software could reenter the design phase or just be marked as complete.

For this stage, the tool used for the development of the software and which performed most of the phases in the SDLC was Unity3D [19]. Conveniently, Unity3D has incorporated an integrated development environment (IDE) to write, compile, and integrate the script. It also allowed the use of external scripts. Visual Studio [20] was taken into account as the IDE for the project in Unity3D. Another important task, within this stage, was the optimization of the models. Unity3D offered the flexibility to work with the lightning and texturing for the several assets imported from other tools. In addition, the scripts have been implemented with the idea to maximize their efficiency in the project by minimizing the code redundancy. This is very important for future support of the software and its extension. The concepts related to object-oriented programming (OOP) are important in the script development, since it is imperative to have a software clear and neat for documentation and suitability.

Unity3D allows scripting in either C# or JavaScript; however, to maintain consistency of the project, the scripts were written entirely in C#. Finally, in this stage, multimedia integration was also implemented. Multimedia resources are included to reinforce the cognitive learning beside the use of the animations and the 3D models. Further, this implementation involves the development of an interface that allows the software to handle multimedia content, such as videos, pictures, and sounds.

Fig. 4 Main menu of the interactive software tool for nursing education



3.3 *Delivery and Testing*

The last stage, of the workflow, covers the delivery and testing phases; however, this overlaps with the tasks from the second stage. Working for the delivery involves the design of the software, because this depends on the feedback given by the user. During this phase, human-computer interaction (HCI) concepts were referred to ensure the development of a friendly graphical user interface (GUI). According to Boeker [21], the usable system implies a system which is easy to learn and easy to utilize, where several factors are important including cognitive link between user and the system, quality of the information, and even dialogues design. One of the main goals is to deliver a software with an intuitive interface, and according to Bartfield [22], this can be reached by the contribution of the following principles: prediction, synthesis, familiarity, and consistency. Following these principles, the interface of the software is made keeping the same design framework along with different menus and modules implemented so far. Figure 4 shows the design of the main menu, which consists of three large buttons for options regarding to the tools and one small button for quitting the program.

In Fig. 5, it is observed how the typography and minimalist design is consistent along the different scenarios of the project. This is expected to reduce the time, for the user, in learning the different features available in the interactive tool. Within this stage, the choice of the platform where the application is going to be used is also analyzed. Initially, the goal was to implement this work as a web application; however, that would require a back-end development to keep track of the results from the final users, database management, and server configuration; therefore, the plan was postponed for a future version of the software. Currently, the development of the software tool is more focused on the front-end aspects, with user experience and educational content. The application has been built as a desktop program for

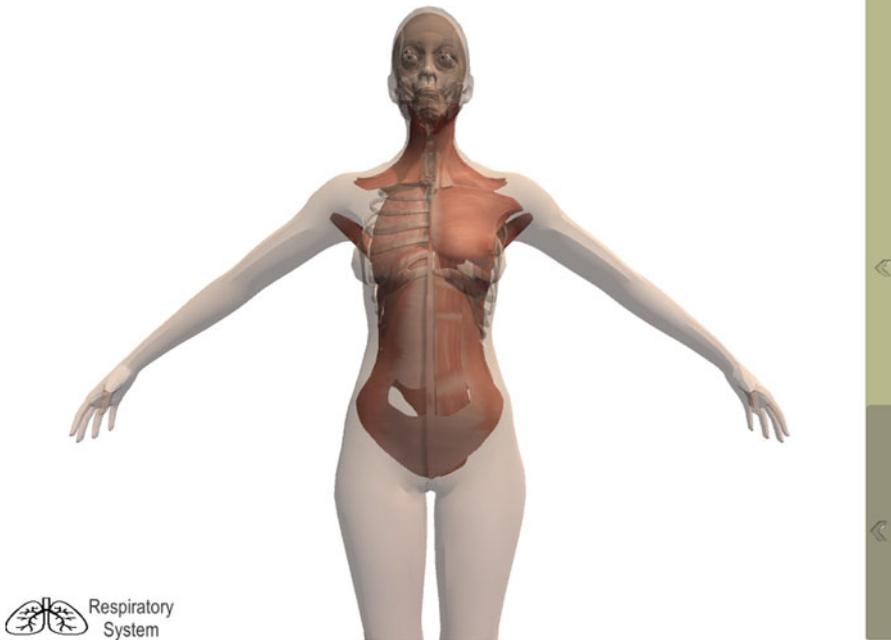


Fig. 5 Peek of one of the modules implemented in the software interactive tool

windows-based operating systems (OS) because it is predominantly used within the university campuses, where our primary target users are enrolled.

The last task, from the workflow, is the testing. The primary test targets are the graduate nursing students enrolled in advanced pathophysiology. The testing phase will consist of two segments: First, a pilot tryout of the software, conducted by the people involved in the development, in front of population belonging to our primary target. A survey will be done to collect feedback from the target population, where the usability and efficiency of the program will be measured in relation to the traditional learning methods. The second segment will come after the review of the feedback. From the feedback, new requirements are expected to be generated along with suggestion for adjustment of the current deliverable version of the software.

4 Pathophysiology Topics

In this section, a summary of the topics to be covered in for the final version of the application will be discussed. In addition, a brief description of the modules already implemented will be provided. Pathophysiology changes from one disease to another and also from one body system to another; for that reason, an efficient

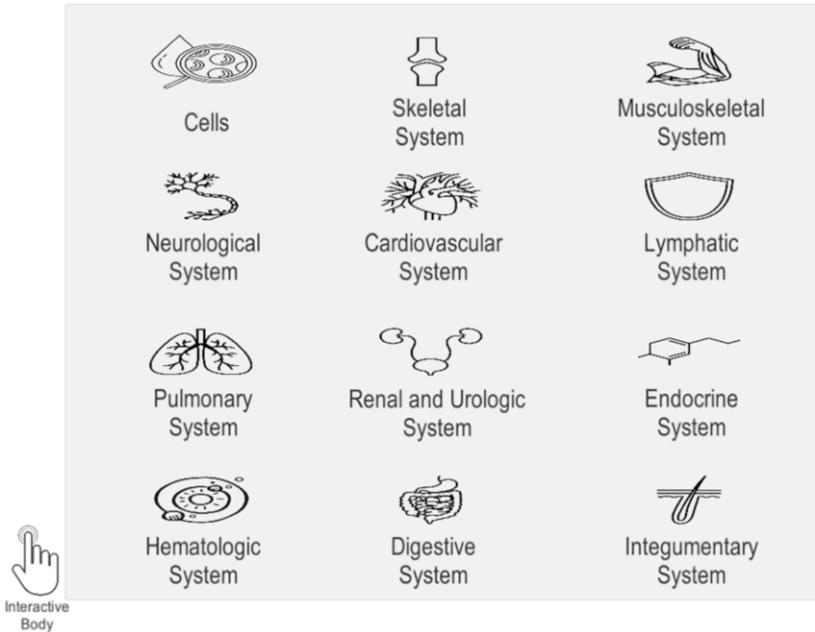


Fig. 6 Human systems covered in the application

way in studying pathophysiology is separating the different diseases and injuries according to the corresponding system of the human body. Our application starts, by showing the different systems covered in the curricula of nursing educational courses as follows.

- Cells
- Skeletal system
- Musculoskeletal system
- Neurological system
- Cardiovascular system
- Pulmonary system
- Renal and urological system
- Endocrine system
- Hematologic system
- Digestive system
- Integumentary system

In Fig. 6, it has been shown how the systems are displayed in user interface of the application. Among the system modules presented in the interface, the project initially is focused on the development of the pulmonary system, by implementing the content for teaching of chronic obstructive pulmonary disease (COPD).

4.1 Pulmonary System

In this module, we developed a sequence of interactive scenarios, where processes related to diseases in the pulmonary system are explained. The initial part in the development of this module was focused on the pathophysiology of COPD. The term COPD is used to describe damage in the airflow related with emphysema and chronic bronchitis [23]. Both disease processes (emphysema and chronic bronchitis) have been explained with the use of several resources in such a way that the cognitive learning and self-learning are achieved. Among the resources used are pictures, videos, audio, animation with the 3D models, capability to interact with the models, and the delivery of text information. All of these contents are presented in four sections for each disease, and those sections are:

- Pathophysiology
- Clinical manifestation
- Treatment
- Risks

Pathophysiology describes the processes related to a particular disease or injury. In the clinical manifestations section, the symptomatology of a particular disease is provided. The treatment section explains the relationship between the pathophysiologic process of the disease and how the treatment disrupts the disease. The treatment section is directly linked to the pathophysiologic processes of disease in such a way to show the effects of different treatments for a particular disease or injury. In Fig. 7, one of the animation sequences in the pathophysiology section for emphysema has been depicted.

4.2 Neurological System

In this module, we are expecting to deliver information regarding to the different diseases that affect the neurological system in the human body. The first disease to be implemented is Parkinson's disease.

Parkinson's disease is a chronic and neurodegenerative disease, characterized by the manifestation of tremor in the hands of the patients afflicted with the disease [24]. For this module, the four sections, pathophysiology, clinical manifestation, treatment, and risk, were implemented ensuring the use of interactivity resources and cognitive learning. The development of most of the animations focuses on how the disease causes progressive neuron degeneration in specific regions of the brain. Figure 8 shows a screenshot of one of the animations describing the degenerative process of neurons due to Parkinson's disease.

5 Conclusion and Future Works

Since the implementation of new technology has been proposed for nursing education, the creation of innovative learning methodologies utilizing e-learning tools is increasing. This paper described an interactive software to learn pathophysiology with 3D virtual models. The development of the software is aligned with the principles of HCI for its GUI such as incorporating an active learning approach. The development is performed under the supervision of researchers from the college of nursing. Currently, the project is within its front-end side development, and the pilot is expected to be tested in the current academic term. Due to the limitation in time, the software has not been pilot tested in the classroom. Based on the feedback received from the pilot, an improved version will be developed as well as the implementation of the remainder systems modules. In the future, we hope to extend the software to a web application in addition to a desktop-based application in order to make it more accessible to users. The design of a back-end implementation will be completed to keep track of the users, have a record of their academic progress through the application, and retrieve feedback to improve the application for future releases.

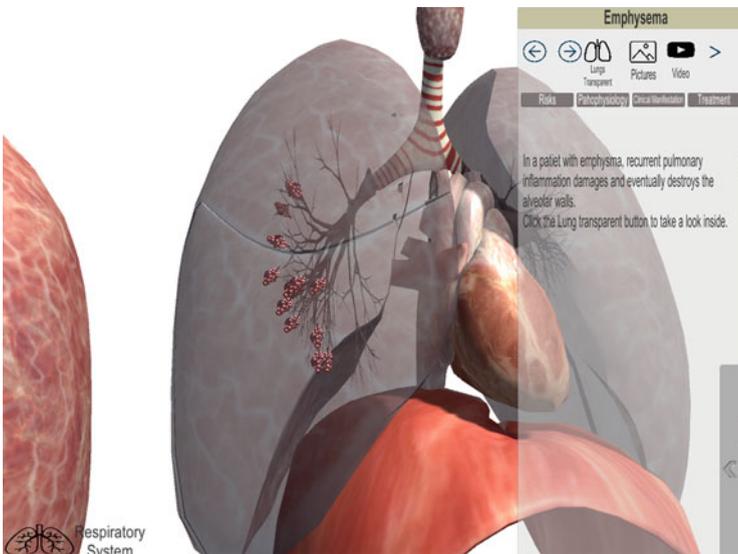


Fig. 7 Screenshot of the pulmonary system module

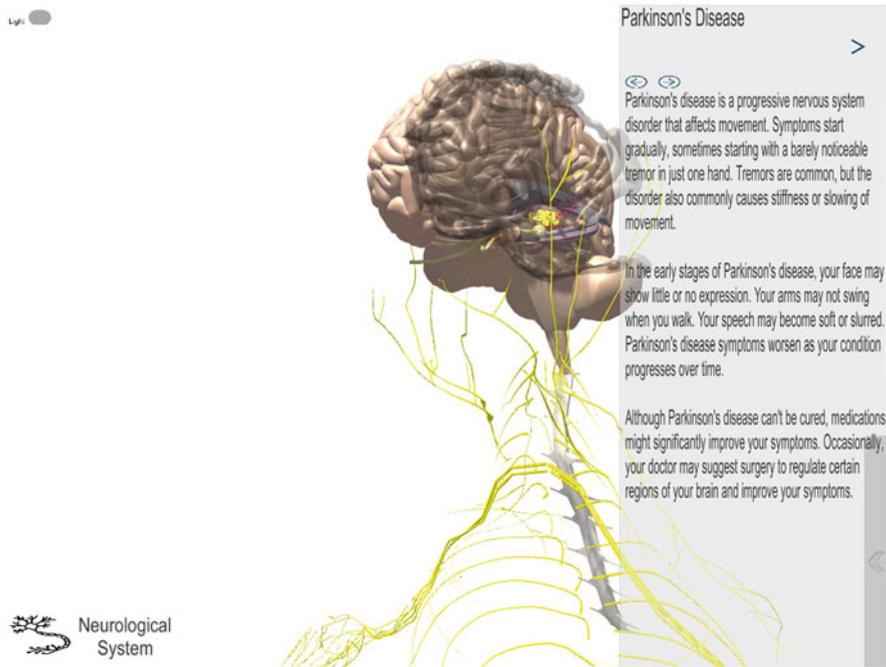


Fig. 8 Screenshot of the neurological system module

References

1. National League of Nursing. Position statement: Innovation in nursing education: A call for reform. <http://www.nln.org/docs/defaultsource/about/archived-position-statements/innovation-in-nursingeducation-a-call-to-reform-pdf.pdf?sfvrsn=4>. Accessed 4 Sept 2019
2. Bonwell, Charles C.; Eison, James A., Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports
3. D.M. Billings, J.A. Halstaed, *Teaching in Nursing: A Guide for Faculty*, 2nd edn. (Elsevier, St. Louis, 2005)
4. National League for Nursing Accrediting Commission. NLNAC accreditation manual. <http://www.nlnac.org/manuals/NLNACManual2008.pdf>. Accessed 4 Sept 2019
5. M.J. Lewis, Computer-assisted learning for teaching anatomy and physiology in subjects allied to medicine. *Med. Teach.* **25**(2), 204–207 (2010)
6. M.O. Forbes, M.T. Hickey, Curriculum reform in baccalaureate nursing education: Review of the literature. *Int. J. Nurs. Educ. Scholarsh.* **6**(1), 1–16 (2009). <https://doi.org/10.2202/1548923X.1797>
7. K.H. Lucas, J.A. Testman, M.N. Hoylans, A.M. Kimble, M.L. Euler, Correlation between active-learning coursework and student retention of core content during advanced pharmacy practice experiences. *Am. J. Pharm. Educ.* **77**(8), 1–6 (2013). <https://doi.org/10.5688/ajpe778171>
8. P. Dias, T. Sousa, J. Parracho, I. Cardoso, A. Monteiro, B.S. Santos, Student projects involving novel interaction with large displays. *IEEE Comput. Graph. Appl.* **34**(2), 80–86 (2014)

9. J.M. Padilha, P.P. Machado, A. Ribeiro, J. Ramos, P. Costa, Clinical virtual simulation in nursing education: Randomized controlled trial. *J. Med. Internet Res.* **21**(3), e11529 (2019). <https://doi.org/10.2196/11529>
10. Wolters Kluwer, 65% of Nursing Education Programs Adopting Virtual Simulation. <http://healthclarity.wolterskluwer.com/nursingeducation-programs-virtual-simulation.html>. Accessed 20 Mar 2020
11. K. Choi, Virtual reality wound care training for clinical nursing education: An initial user study. 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan, 2019, pp. 882–883. doi: <https://doi.org/10.1109/VR.2019.8797741>
12. T. Harjanto, D.S.E.W. Sumunar, K.A.K. Putri, E-Learning implementation on clinical rotation nursing education: A case report of Universitas Gadjah Mada. 2018 4th International Conference on Science and Technology (ICST), Yogyakarta, 2018, pp. 1–5. doi: <https://doi.org/10.1109/ICSTC.2018.8528612>
13. C.F. Waltz, L.S. Jenkins, N. Han, The use and effectiveness of active learning methods in nursing and health professions education: A literature review. *Nurs. Educ. Perspect.* **35**(6), 392–400 (2014). <https://doi.org/10.5480/13-1168>
14. I. Gorbanev, S. Agudelo-Londono, R.A. González, A. Cortes, A. Pomares, V. Delgadillo, O. Munoz, A systematic review of serious games in medical education: Quality of evidence and pedagogical strategy. *Med. Educ. Online* **23**(1), 1–9 (2018). <https://doi.org/10.1080/10872981.2018.1438718>
15. D.R. Krathwohl, A revision of Bloom’s Taxonomy: An overview. *Theory Pract* **41**(4), 212–218, <https://www.depauw.edu/files/resources/krathwohl.pdf>. Accessed 30 Sept 2019
16. M. Boeker, P. Andel, W. Vach, A. Frankenschmidt, Game-based elearning is more effective than a conventional instructional method: A randomized controlled trial with third-year medical students. *PLoS ONE* **8**(12) (2013). <https://doi.org/10.1371/journal.pone.0082328>
17. Autodesk Maya. [Online]. Available at: <https://www.autodesk.com/products/maya/overview>. Accessed 19 Mar 2020
18. SDLC Models — Software Development Life Cycle Models. [Online]. Available at: <https://www.learntek.org/blog/sdlc-models-softwaredevelopment-life-cycle-models/>. Accessed 19 Mar 2020
19. B. Shneiderman. Unity. [Online]. Available at: <https://www.unity.com>. Web Accessed 19 Mar 2020
20. Visual Studio. [Online]. Available at: <https://visualstudio.microsoft.com/>. Accessed 19 Mar 2020
21. M. Boeker, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 3rd edn. (Addison-Wesley Pub Co, 1997)
22. L. Bartfield, *The User Interface* (Addison-Wesley, Concepts & Design, 1993)
23. *Anatomy & Pathology*, 4th edition, Lippincott Williams & Wilkins, 2005, p. 40
24. G. DeMaagd, A. Philip, Parkinson’s disease and its management: Part 1: Disease entity, risk factors, pathophysiology, clinical presentation, and diagnosis. *P T* **40**(8), 504–532 (2015)

A Simulation-Optimization Technique for Service Level Analysis in Conjunction with Reorder Point Estimation and Lead-Time Consideration: A Case Study in Sea Port



Mohammad Arani, Saeed Abdolmaleki, Maryam Maleki,
Mohsen Momenitabar, and Xian Liu

1 Introduction

An international operating port with no intermission and high imposed cost due to stops caused by lack of equipment and improper inventory system to maintain the essential functionality of machines, challenging inventory monitoring owing to an immense diversity of items and the varied significance level of items, increased the awareness of managers about the service level analysis of the inventory items of Shahid Rajaee Container Terminal, Iran. This scientific and practical approach offers a step-by-step procedure to be taken by similar medium- and large-sized industries as a guideline to successfully establish an effective inventory system as following three stages:

1. Stage one

- (a) Identifying the inventory items (variety and consumption rate)
- (b) ABC analysis (combining with the AHP method)

M. Arani (✉) · M. Maleki · X. Liu

Department of Systems Engineering, The University of Arkansas at Little Rock, Little Rock, AR, USA

e-mail: mjarani@ualr.edu; mmaleki@ualr.edu; xxliu@ualr.edu

S. Abdolmaleki

Department of Industrial Management, The University of Shahid Beheshti, Tehran, Iran

M. Momenitabar

Department of Transportations and Logistics, The North Dakota State University, Fargo, ND, USA

e-mail: mohsen.momenitabar@ndsu.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_61

839

- (c) Scoring inventory items in terms of importance and viability to apply advance inventory control methods
- (d) Selecting the main items according to their score

2. Stage two

- (a) Identifying the demand distribution function of the main items
- (b) Identifying the distribution function of their lead times
- (c) Identifying the inventory control costs (including holding, order, and shortage cost) for the main items

3. Stage three

- (a) Identifying the expenses for procurement and supply of the main items
- (b) Simulation of the inventory control system and its analysis
- (c) Determining the optimal inventory policy for the main items (ROP and its quantity)
- (d) Service level analysis

Therefore, this original research, in one point of view, walks the enthusiastic readers through each step and resolves technical halts she/he might encounter in inventory monitoring. Alternatively, the novel approach comprehends several contributions that distinguish this study from its counterparts: (a) the way it deals with extraordinarily huge data set requires to be cleaned and processed with regard to the importance of each item; (b) embedded simulation-optimization technique and employed roulette wheel concept to derive the demand quantity for the main inventory items, (c) not only the real data set but also the technical expertise verification of the final simulation model and catered analysis, are valuable assets to this academic and practical research.

In the following, first, we provide relevant and state-of-the-art literature thus far. In Sect. 3, the problem statement is described. Solution methodology encompassing the mentioned three steps is in Sect. 4. In the next section, service level analysis is individually furnished as one of the advanced technical analysis tools. Finally, to conclude the study, managerial insights with a concentration on inventory management are suggested.

2 Literature

Inventory items (for instance, raw materials, semi-finished products, finished products, and spare parts) are among the assets that require acute control and efficient management. There are compelling grounds to store inventories; (a) to prevent interruptions in production operations; (b) to prevent bottlenecks in operation; (c) to timely produce and deliver, accompanying with quality customer services; (d) to operate in the same level, working-station-wise; and (e) to protect against shortage and price fluctuation in raw materials. Therefore, many companies are ought to store

inventory to swiftly deal with these prevailing conditions and maintain not only an acceptable service level but also a minimal inventory cost. A study of well-established companies' balance sheets shows that about 15–35% of companies' capital comprehends inventories. It also demonstrates that approximately 17–24% of the monetary value of inventories is spent on holding costs; furthermore, the inventory-to-sales ratio is about 12–20%, and the inventory-to-asset ratio is approximately 16–30% [1].

Since the paramount importance of inventory control is clearly articulated, the concerned case study of this research is not an exception bearing in mind the 24,700 items, likewise. One definition of service level that could be modified for our research is obtained from [2], decently defined as “the probability to meet demand while an order is placed.” The main focus of the research, however, is to examine the effect of safety levels on δ -service level which refers to “the probability to meet a demand with a substitute item while an order is placed.” One prior study that stands out among the scholarly works, in the sense of similarities, is conducted by [3] on the subject of fashion supply chain inventory. The optimal inventory model is investigated, while the service level and lead time are controllable. The authors proposed two inventory models responding to buyer's and vendor's inventory behavior. In our approach, however, the optimal inventory model is gained by simulation-based optimization modeling considering that the lead time is dictated by the vendor's historical data and a range of service level studied to cater DM with options associated with incurred cost on the entire system of inventory. Although the definition of service level seems wide range, Candas and Kutanoglu [4] employed customer-based service levels for their inventory model integrated with a location-allocation problem. Transchel and Hansen [5] proposed an inventory model considering service level constraints and uncertainty in lead time, while the cornerstone is to take into account the perishability of products tackled by simulation-based optimization method. As elucidated, the perishability of products refers to having either finite shelf life or physical quality decay over time.

The simulation approach is an extensively employed method used to model the complex inventory systems and diverse inventory policies. In one study that combines simulation analysis with big data in retail environment proposed by [6]. The provided case study observed an automated refreshment system in Japan with 3000 items that were continually updated by real POS data. The simulation-based algorithm led to a lower inventory level and maintained the service level at the same time. Reference [7] also proposed a simulation-based optimization method called “sample average approximation” in which the service level is treated as a constraint. Kim and Jeong [8] combined a time series analysis – called ARIMA in order to predict the demand in accordance with the EOQ model – and a simulation model to measure the effectiveness of the model. Along with the previous paper, [9] employed a simulation-based model to determine the safety stock level using Arena software. In their study, they selected only three products to investigate the relationship between the safety stock level and service level. One novel research paper using Vensim software [10] proposed a system dynamic simulation approach

to improve inventory performance in the consumer products trade business unit. According to the obtained results, proposed modifications resulted in 43,000 dollars saving of operational cost per year. To conclude this paragraph on the simulation technique to deal with the inventory problems, please refer to these recent papers as well [11–13].

As clarified earlier, one segment of this research and the scientific trend is to use varied ABC methods to classify inventory items. Eraslan and Tansel IC [14] proposed an improved decision support system (IDSS) for inventory classification utilizing the ABC method. In the developed software, the proper ABC classification model is selected among five approaches, for instance, the analytic hierarchy process (AHP). One of the close research papers to ours is offered by [15] in which the authors hired an AHP model to classify the spare parts inventory management for the aviation industry. The case study is an aircraft maintenance site in Indonesia intended to reduce unnecessary site stop times. The authors claim that after the validation phase, the model proved to be accurate and rapid in response. In addition to the case studies observed in the literature, one is offered by [16]. In this study with real-world data, an inventory model is provided for the chassis parts at the US container ports. The mentioned study provides a mathematical model that could be considered as a decision support system for the planning of neutral chassis.

3 Problem Statement

Inventory management plays a critical role in any business, whether manufacturing or services, and with proper control, one significant step is taken to balance the flow of operations. Inventory control is a common delicate matter that all kinds of companies need to cope with. Moreover, how to handle the mentioned matter is the difference between successful companies and unfortunate ones. In developing countries, the capital stored in the form of inventories is usually higher in comparison with developed countries [17]. Therefore, an accurate estimation of ROP and the quantity of order prevent accumulating and freezing an enormous amount of money in form of inventory items, or system breakdown in terms of disruption in manufacturing or for services rendered, otherwise.

Here, inventory cost entails three well-known parts, namely, holding cost, order cost, and shortage cost which is considered in the model. In the following, Fig. 1, a schematic design of the ROP model is demonstrated.

4 Methodology

In general, this research has the following features: (a) it is categorized into a practical study with its technical contribution; (b) in terms of data collection method, it is descriptive; (c) in terms of analysis point of view, it is a real-world case

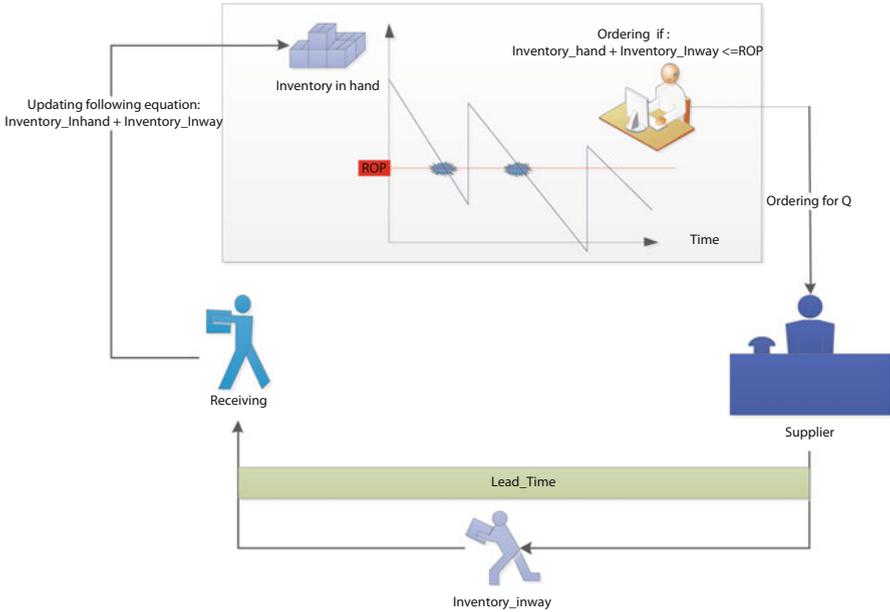


Fig. 1 Conceptual model

study; and (d) in terms of solution approach, it is quantitative-analytical (operations research) which has a foundation on mathematical modeling.

4.1 Step 1: ABC

By properly controlling inventory and planning, you can take steps to balance the flow of operations and reduce the overall production cost. In inventory, a small number of items usually account for the bulk of the inventory’s value, and vice versa, a large number of items have a small monetary value. Therefore, having a single inventory control method does not seem reasonably sound for all of these items. In this successful technical project, the ABC analysis is used to identify items whose inventory control is more effectively beneficial in the long run. In addition to basic criteria such as consumption volume and purchase price, other tangible factors influencing the classification of items are also considered. In Pareto’s analysis method, which is well celebrated and recognized as ABC classification, the items are divided into three groups (or more) A, B, and C, as follows: group A includes goods with a high monetary value (or consumption), which is a small percentage; group B includes goods with an average monetary value, which also includes an average percentage; and finally, group C contains goods with low monetary value but a high percentage. The result of such an analysis shows that we must strictly control class A, control class B less, and control class C less than the first two classes (Fig. 2).

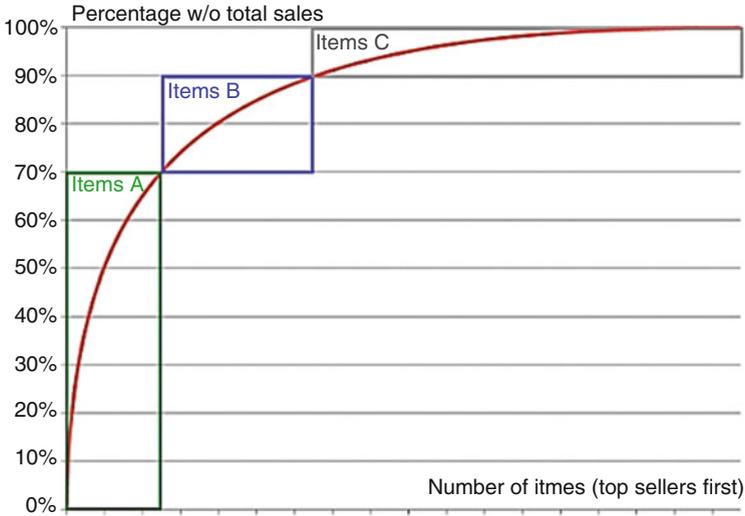


Fig. 2 ABC classification

The purpose of this method is to employ the coefficient of importance or necessity of items, which is obtained by questionnaire and interview with experts and managers of the company which led us to determine the true value of items not only regarding their monetary values, merely. Finally, the relative value of all items is prepared, and based on that, a prioritization is determined for them. The steps for inventory items classification into high, medium, and low value are as follows:

1. Make a list of the total annual value of all items stored in the inventory and sort them by the highest total value.
2. Assign a degree of value (importance or necessity) for each item (the largest value will be 1).
3. Prepare a table of total values.
4. Calculate the ratio of the total annual value of each item in inventory.

In Fig. 2, the horizontal axis shows the number of inventory items, and the vertical axis shows the percentage of the class's value.

By considering several ABC analyses (including Figs. 3 and 4) and according to the Pareto chart, the consumption value to inventory level (Fig. 4) was selected as the best method of Pareto analysis of inventory items. On the other hand, preliminary studies have shown that many of the items in the warehouses are stagnant or the available data are not accurate enough.

Fig. 3 Inventory value to the inventory level

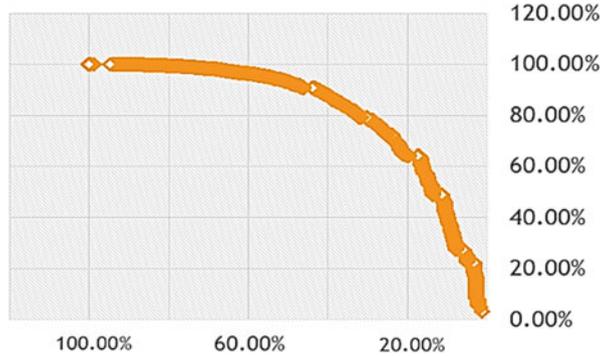
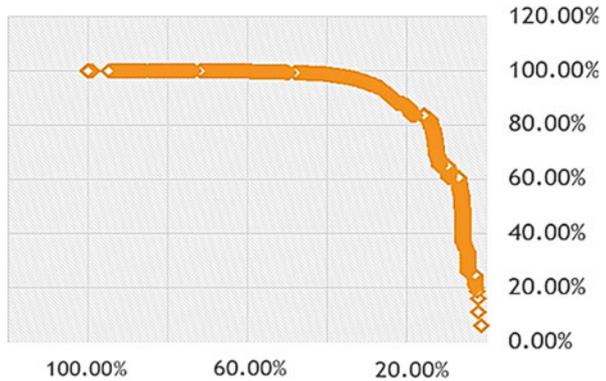


Fig. 4 Consumption value to the inventory level



4.2 Forecasting

The most common possible models, which are commonly used from two different approaches to predicting inventory systems, are Poisson and exponential distribution functions. In the Poisson probability mass function, λ represents the average number of Kanban orders in the time interval $[0, t]$.

$$P [N(t) = n] = \frac{e^{-\lambda t} (\lambda t)^n}{n!}, \quad t \geq 0, \quad n = 1, 2, \dots \tag{1}$$

When events occur over time randomly and based on a Poisson process with a λ rate, then the time required for an event to occur will have a $1/\lambda$ parameter of the exponential probability density function (PDF).

$$f(x) = \frac{1}{\lambda} e^{-\frac{x}{\lambda}}, \quad x > 0, \quad \lambda > 0 \tag{2}$$

By identifying the unknown parameters introduced in the above equations, we can use the probability density functions and various software, such as Arena and Minitab, in order to predict the demand and waiting time for inventory items in the future periods.

4.3 Step 2: AHP

One of the most effective decision-making techniques is the AHP method proposed by Thomas L. Saaty in the 1970s. This technique is based on pairwise comparisons and allows managers to study different scenarios. The AHP has been received warm welcomes by various managers and users due to its simplicity yet comprehensive nature. This technique makes it possible to formulate the problem hierarchically, and besides, it is possible to consider different quantitative and qualitative criteria in the problem. This process involves a variety of options in decision-making and allows for the analysis of sensitivity for criteria and sub-criteria. The method is also based on pairwise comparisons that facilitate comparison and calculation. Also, it can show the degree of compatibility of the decision, which is one of the excellent advantages of this technique in multi-criteria decision-making. Saaty has stated the following four principles as the principles of the hierarchical analysis process and has based all calculation rules on these principles. These principles include (a) reciprocal condition, (b) homogeneity, (c) dependency, and (d) expectations.

4.3.1 Weights Calculation

The AHP analyzes complex problems, transforms them into simple forms, and solves them. The following steps must be taken to resolve an issue or decision [18]:

- (a) Model the problem in a hierarchical format encompassing the objective, alternatives, and criteria for evaluating the alternatives.
- (b) Set priorities among the elements of the hierarchy by making a judgment in a row based on pairwise comparisons of the factors.
- (c) Combine the series of judgment to conclude a set of priorities for the hierarchy.
- (d) Control the consistency of the judgment according to the four mentioned principles.
- (e) Eventually, reach a final decision.

The calculation of weight in the hierarchical analysis process is discussed in two separate sections: (a) relative weight and (b) absolute weight. The relative weight is obtained from the pairwise comparison matrix, while the absolute weight is the final rank of each option, which is calculated from the combination of relative weights.

(a) Relative weight

In the AHP, the elements are first compared in pairs and the matrix of pairwise comparisons is formed. Then, using this matrix, the relative weight of the elements is calculated. In general, a pairwise comparison matrix, A , is shown as follows, in which the a_{ij} is the preference of the element i over the element j ; therefore, the weight of the element, w_i , is obtained according to the value of a_{ij} .

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \tag{3}$$

Any pairwise matrix, A , may be consistent or inconsistent. When this matrix is consistent, the calculation of the weights (w_i) is simple and is obtained by normalizing the elements of each column. However, in a case where the matrix is inconsistent, it is not easy to calculate the weights, and there are four main methods for obtaining it, which are (a) least squares method, (b) logarithmic least squares method, (c) eigenvector method, and (d) approximation methods. It should be noted that since the first three methods have heavy computations, some approximate methods have been proposed that are less accurate and have less and simpler calculations. These methods are mainly approximations of the eigenvector method that correct the calculations with different accuracy.

(b) Absolute weight

The final weight of each option is obtained from the sum of the multiplication of the value of the criteria and the weight of the option.

4.3.2 Consistency Calculation

As mentioned earlier, a matrix may or may not be consistent. In a consistent matrix, the calculation of weight is simple and is achieved by normalizing a single column, while several methods have been mentioned for calculating weights in the inconsistent matrix. Calculating the level of inconsistency is very important because it shows how much confidence can be given to priorities from comparisons. In general, it can be said that the acceptable level of the inconsistency of a matrix depends on the decision-maker, but Saaty presented the number 0.1 as an acceptable limit and believed that if the level of inconsistency is more than 0.1, it is better to reconsider the judgment.

(a) Consistent matrix

If there are n criteria, namely, c_1, c_2, \dots, c_n , the pairwise comparison matrix is as follows:

$$A = [a_{ij}]; \quad i, j = 1, 2, \dots, n \tag{4}$$

where a_{ij} shows the preference of the element c_i over c_j if the following holds:

$$a_{ik} \times a_{kj} = a_{ij}; \quad i, j, k = 1, 2, \dots, n \tag{5}$$

Then, A is a consistent matrix.

(b) Inconsistent matrix

In this section, we want to know if the matrix of the pairwise comparison is inconsistent, what is the degree of inconsistency of the matrix, and how do we measure it. Before stating the criteria for measuring matrix inconsistency, the following are three important theorems about any matrix of pairwise comparison:

Theorem 1 If $\lambda_1, \lambda_2, \dots, \lambda_n$ are the eigenvalues of matrix A , the sum of all eigenvalues equals n .

$$\sum_{i=1}^n \lambda_i = n \tag{6}$$

Theorem 2 The largest eigenvalue (λ_{max}) is always greater than or equal to n (in which case some λ will be negative):

$$\lambda_{max} \geq n \tag{7}$$

Theorem 3 If the matrix elements deviate a small distance from the consistency state, the eigenvalues will also deviate a small distance from their consistency state.

$$A \cdot w = \lambda \cdot w \tag{8}$$

where λ and w are the eigenvalues and the eigenvector of matrix A , respectively. When matrix A is consistent, a special value is equal to n (the largest eigenvalue) and the rest is zero. Therefore, in this case, the following could be stated:

$$A \cdot w = n \cdot w \tag{9}$$

If the matrix of A is inconsistent, according to Theorem 3, λ_{max} moves slightly away from n , so we have:

$$A \cdot w = \lambda_{max} \cdot w \tag{10}$$

The reason to use λ_{max} according to Theorem 3 is that it will be the shortest distance from n . Since λ_{max} is always greater than or equal to n , and if the matrix deviates slightly from the consistency state, λ_{max} will deviate slightly from n , so the difference between λ_{max} and n ($n - \lambda_{max}$) depends on the value of n , and to eliminate this dependency, an index can be calculated as the following. The index is the definition of the inconsistency index (II):

$$II = \frac{\lambda_{max} - n}{n - 1} \tag{11}$$

The values of the inconsistency index (II) are calculated for matrices whose numbers are completely random, and it is called the random matrix inconsistency index (RMII); the values for the following n matrices are as follows (Table 1):

For each matrix, the result of dividing the inconsistency index (II) by the random matrix inconsistency index (RMII) is a proper criterion for making a judgment about inconsistency, which is called the inconsistency rate (IR). If this number is less than or equal to 0.1, the system’s inconsistency is acceptable; otherwise, the process must be reconsidered. Finally, according to the mentioned procedure, the AHP method is performed with the assistance of Expert Choice software. In the following, the final five criteria – among 27 well-known ones in literature studies – to classify inventory items were selected by experts’ opinion:

- (a) Critical degree: Critical goods or materials are those items that are considered critical by the company for reasons related to their consumption or purchase process, and its deficiency can be detrimental.
- (b) Item consumption: There are different parts of the site where the items are consumed.
- (c) Lead time: The time between ordering and receiving the goods in stock.
- (d) Availability: Possibility to provide, purchase, and transfer items quickly to the warehouse due to the abundance of items or the availability of supply contracts.
- (e) Inventory turnover: In accounting, it is a ratio, which indicates that the company’s inventory of items and materials, in a certain period of time (such as in one fiscal year), has been consumed or sold several times and replaced.

Using Expert Choice software, we obtained the weight of each criterion. The weight values for the criteria are in Table 2.

4.4 Calculate the Qualitative and Multivariate Values of Items Using the AHP Concept

To classify items using the AHP method, we first normalize the values of the items in each criterion. The normalized value of each item for each criterion was obtained based on the common procedure in the AHP method linearly and by dividing each of the values by the sum of the corresponding standard columns. The method of calculating the relative rank of each item can be expressed as follows:

Table 1 Random matrix inconsistency index (R.M.I.I.)

n	1	2	3	4	5	6	7	8	9
R.M.I.I.	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45

Table 2 Final criteria weights

Criteria	(a)	(b)	(c)	(d)	(e)
Weight	0.52	0.15	0.14	0.12	0.7

$$R_i = \sum_{j=1}^5 W_j \cdot V_{ij} \quad (12)$$

where

W_j is the relative weight of criterion j

V_{ij} is the normalized value for item i regarding criterion j

After calculating the qualitative value for all main items, the consumption value was obtained by multiplying the consumption amount by the price. Due to the fact that the qualitative values obtained for the items were normalized and to combine quantitative and qualitative values, these quantitative values of consumption were also normalized by adding any value to the total quantitative value of the goods. In order to combine quantitative and qualitative values for each item and to calculate the combined value, it was necessary to consider a weight for each of these two types of values. Due to the involvement of several criteria in calculating the qualitative value, an attempt was made to consider a higher weight for the qualitative value against the quantitative value of the sole quantitative criterion. After performing different scenarios and finally according to the quality of the Pareto diagram obtained for each scenario, the ratio of 6 to 1 was selected for qualitative value compared to quantitative value. Therefore, how to calculate the combined value of each product can be expressed as follows:

$$G_i = \frac{6}{7}R_i + \frac{1}{7}K_i \quad (13)$$

where

R_i is the qualitative value of several criteria obtained for item i

K_i is the quantitative value of the amount of consumption obtained for item i

G_i is the combined value obtained for item i

After calculating the combined value for all items, the initial classification of goods was done by the ABC classification method. In this classification, the Pareto diagram of items to determine items in classes A, B, and C was obtained by comparing the value of inventory with the number of items in the inventory, which is as follows (Fig. 5). The vertical axis is the cumulative percentage of integrated values, and the horizontal axis is the cumulative percentage of inventory. Finally, 1416 items were selected for simulation analysis with Arena software.

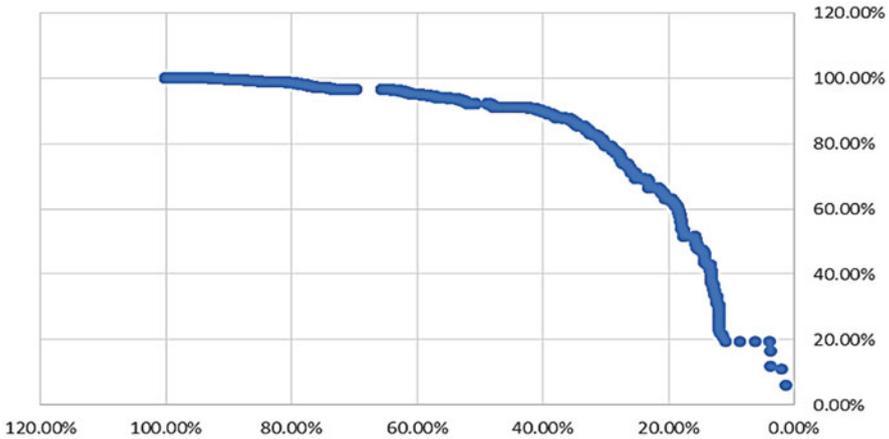


Fig. 5 Integrated ABC classification with AHP

4.5 Step 3: Simulation

4.5.1 Extrapolation

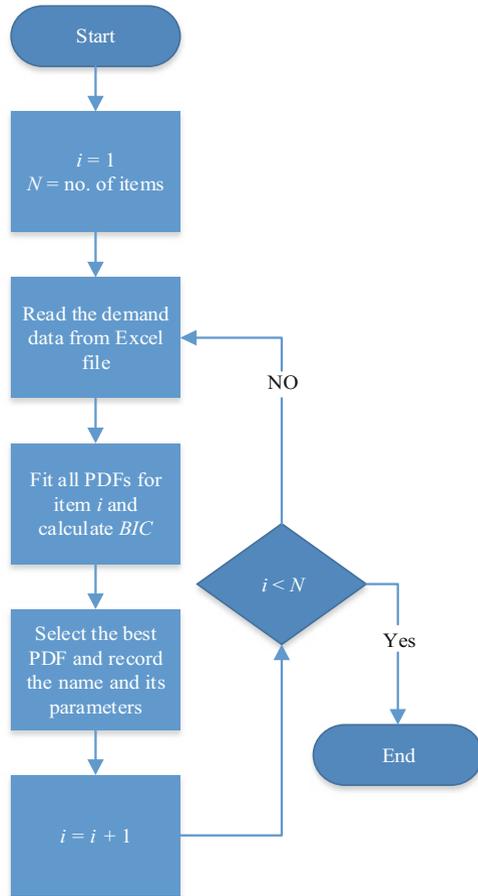
The first step to derive the required information for initializing simulation is to conduct data gathering. The most important source of information and analytical basis for identifying the distribution function of demand and lead time is the historical information that can be extracted from the existing inventory system and documents or through the opinion of the company’s experts. In addition to historical information, the company’s development plans and estimation of future trends can also be considered.

The first analysis that needs to be done to model and simulate 1416 main items of the inventory is to identify their distribution function (fit the probability density function). For this purpose, according to the list of probability density functions available in Arena software, the fit algorithm was implemented in the MATLAB software environment. The reason for using MATLAB software is the higher ability and accuracy of this software in fitting probability density function. After identifying the most appropriate function for each item, the type of function and its parameters are entered into the simulation model in Arena as required inputs. In this project, the Bayesian information criterion (BIC) is used to select the best distribution for data of demand. The index is calculated as follows:

$$BIC = k \times \ln(n) - 2 \times \ln(L) \tag{14}$$

where k is the number of parameters to be estimated, n is the number of observations, and L is the maximum value of the correction function for the model. Among the candidate distribution functions, the one with the lowest BIC level is the best fit. The

Fig. 6 The procedure used to fit the distribution function



procedure used to fit the distributions in this project is in the form of a flowchart shown in Fig. 6.

4.5.2 Implementation

In order to establish a simulation model, the following steps must be taken:

1. Determining the boundary of the studied system
2. Making a model in the software (here Arena 14.0)
3. Determining the nature of the model in terms of a finite or a non-finite system (here the inventory control system is classified as a non-finite system [19])
4. Confirmation and validation of the model implementation in accordance with the performance of the real-world model

5. Discussion and reports

The inventory control model in this software consists of three subsections. Moreover, the demand period is annual.

The *first part* deals with demand, in which the following assumptions are considered:

1. Demand enters the system at the end of each year.
2. The amount of demand per month is proportional to the annual demand for inventory items in the last 9 years.
3. With each entry of demand into the model, the amount of inventory accumulated with demand is examined and questioned, while there are two answers:
 - (a) The demand can be met by the available inventory (in-hand inventory).
 - (b) The amount of inventory is not sufficient to fully satisfy the demand; in other words, part or all of the demand cannot be satisfied; in this case, the amount of unsatisfied demand is considered as a shortage.

The *second part* is related to inventory supply system and inventory monitoring; here the ROP-ROQ method is used. The following assumptions are also used in this section.

1. The inventory level is monitored online and continuously so that after each order arrival, the total inventory level (in-hand inventory) and the coming items (on-way inventory) are less than or equal to the ordering point (ROP) before a new order is placed.
2. With the occurrence of the order, the system status changes to be ready to monitor the inventory level.
3. When the time for receiving the orders arrives, they are immediately added to the inventory of the warehouse and change the status to the ready-to-use state; that is, the amount of inventory on the way is added to the inventory in hand. And for each order, the corresponding order cost is considered.

The *third part* has two subsections:

1. Recording holding cost:

The holding cost period is considered to be annual, for which purpose a separate part of the model is responsible for updating the holding costs. At the end of each year, the amount of remaining inventory in the warehouse is multiplied by the holding costs unit and recorded.

2. Record all costs, including holding, ordering, and shortage costs:

In order to collect all the costs imposed on the system, there is a section in Arena software that allows the user to collect the costs at the end of the simulation period (end of simulation clock). This section is called the statistical data module.

4.5.3 General Simulation Process (Arena): Optimization (MATLAB)

After constructing, reviewing, and approving the inventory system simulation model, in a general process, a simulation-optimization method of this system was performed. This general process, which was simulated in Arena software and optimized in MATLAB software, is accompanied by the VBA interface and illustrated as follows (Fig. 7).

Roulette Wheel

To simulate the inventory model, it is necessary to predict the amount of consumption in different periods, as well as extracting the history of consuming items without considering overhaul from the database, and get the correct pattern of demand behavior. The prediction model is as follows (Fig. 8):

Using annual consumption data of spare parts, we form a distribution function. The distribution function indicates the chance of occurrence of each of the consumption ranges. For example, in Fig. 8, section A, the probability that the amount of spare parts used in the first interval is more likely than the other intervals (the probability of occurrence is 4/11, while for the other intervals, it is 1/11, 2/11, 3/11, and 1/11, respectively). Now, using these possibilities, we use the roulette wheel technique. The roulette wheel is such that the more probable the range, the larger

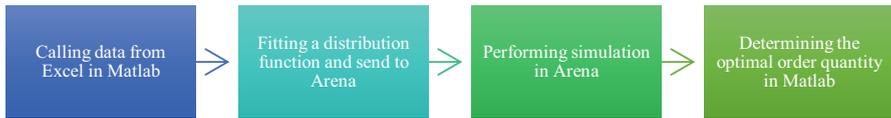


Fig. 7 Simulation-optimization procedure

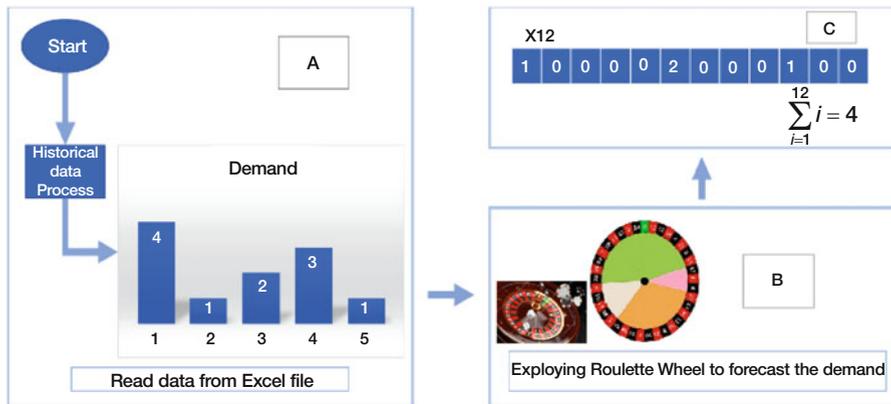


Fig. 8 Roulette wheel

the area of the wheel. As shown in Fig. 8, part B, the green part indicates the highest chance of occurrence. Now, if we rotate the roulette wheel, the indicator will be selected in each interval that stands, and the amount of the annual consumption prediction in that interval will be determined.

Because the simulation accuracy is higher per month, the annual consumption should be converted to monthly. This is done using a uniform distribution so that each month of the year has a chance to be equal in consumption. Also, the amount of consumption in months should be such that it is equal to the total annual forecast. As shown in Fig. 8, section C is first randomly selected for the first, sixth, and tenth months, and then the corresponding consumption values (1, 2, 1, respectively) are assigned to the accident.

5 Service Level Analysis

It should be noted that the service level means the possible percentage of meeting the demand for items, and it is a function of ROP value. On the other hand, the specified ROP value determines the amount of capital expenditure according to the holding cost of items. By service level analysis, we provide the fluctuation in total inventory cost. Fundamentally, by changes in service level and inventory level, an analysis of cost is provided.

Calculation Logic The level of service measures the probability of a shortage. Therefore, the order point must be defined in such a way that it can respond to the demand after the order is issued until the items are received, that is, lead time. Therefore, the distribution function of the occurrence probability of demand at the waiting time must be recognized to obtain the probability of shortage according to the reorder point. After obtaining the demand distribution at the waiting time, according to the desired service level, a reorder point is computed. In the following, the computation steps for the reorder point are explained:

1. Recognize the demand distribution based on historical data.
2. Demand distribution is obtained for the lead time. Mean and variance are obtained for the lead time according to the following equations (in the following equation, instead of a , we plug in the lead time):

$$E(aX) = a(E(x)); \quad \text{Var}(ax) = a^2\text{Var}(x) \quad (15)$$

3. According to the obtained mean and variance, the distribution parameters are corrected and the demand distribution is obtained for the lead time.
4. Depending on the service level of α , the reorder point is obtained through the following relationship:

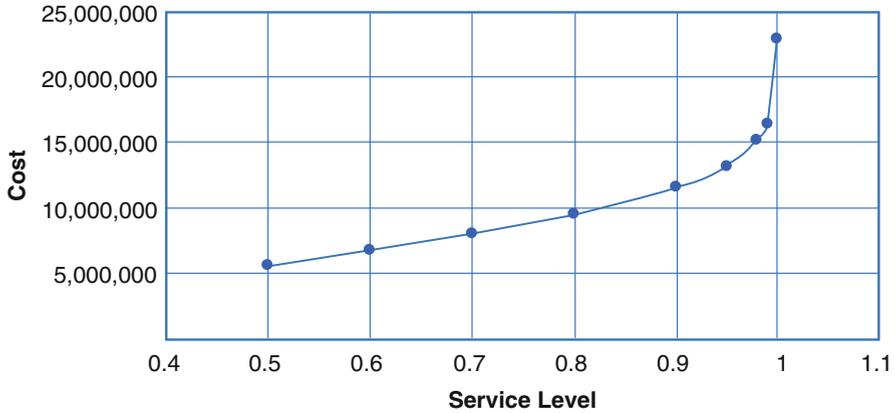


Fig. 9 Service level – cost analysis

$$ROP = F^{-1}(\alpha) \tag{16}$$

Based on the above procedure and in order to draw the relationship between service level and inventory costs, the total inventory cost for the following service levels 50%, 70%, 80%, 90%, 95%, 98%, 99%, and 99.99% calculated and the obtained results are shown in Fig. 9. As can be observed in Fig. 9, the higher the service level, the higher the overall inventory cost will be. In other words, the additional investment on the inventory system while changing the service level from 0.8 to 0.9 is much higher than adapting the service level from 0.6 to 0.7.

6 Conclusion

In this paper, we aimed at proposing a technical and novel approach to properly monitor the inventory system for the main items. Firstly, an ABC classification is offered, and then, with the assistant of the AHP methodology, an integrated approach of qualitative and quantitative methods led us to an applicable ABC classification. While the classification is made, the class A consisting of 1416 items were identified as the main items of the inventory worthy of strict control. Finally, a simulation optimization method determines the best reorder point and the quantity of each order in the recognized inventory policy reorder point and reorder quantity model. The following suggestions are provided which is founded on practical experience and close observation of the system:

- Implementing new parts supply methods, such as vendor-managed inventory (VMI) to reduce or eliminate the inventory of low consumption and stagnant items.
- Diagnosis of the processes of supplying low consumption and stagnant parts.
- Considering an effective integration of maintenance and repair processes pertinent to the inventory items.
- It is suggested that appropriate key performance indicators (KPIs) need to be calculated from the perspective of efficiency and optimization of operating costs, such as inventory cost in detail and inventory turnover.
- In the case of non-stagnant items in class B of Pareto classification, mixed inventory policies are proposed such as periodic inspections.

To address future pathways for this technical paper, we would like to suggest two approaches. Firstly, fuzzy techniques could be a perfect complement to the ABC method since it allows considering quantitative and qualitative criteria simultaneously. The fundamentals could be actively sought in [20, 21]. Secondly, heuristics and meta-heuristics algorithms could be an alternative to combine ABC classification with, for instance, consideration of hybrid ABC-GA approach for deteriorating items in inventory by [22].

Acknowledgments We would like to cherish the moment and sincerely thank the managers of Shahid Rajaee Container Terminal in Iran who ardently support this non-profit academic project by sharing insensitive data with us. This paper is the fruit of the final academic project involving collaboration with industry at the University of Shahid Beheshti in Tehran, Iran.

References

1. S.M.T. Fatemi Ghomi, *Planning, production control, and inventories*, 9th edn. (Amir Kabir, Tehran, 2015)
2. M. Buisman, R. Haijema, E.M.T. Hendrix, On the δ -service level for demand substitution in inventory control. *IFAC-PapersOnLine* **51**(11), 963–967 (2018). <https://doi.org/10.1016/j.ifacol.2018.08.483>
3. G. Li, Y. Kang, and X. Guan, Fashion Supply Chain Inventory Optimization Models with Service Level and Lead Time Considerations, 2016, pp. 237–249
4. M.F. Candas, E. Kutanoglu, Integrated location and inventory planning in service parts logistics with customer-based service levels. *Eur. J. Oper. Res.* **285**(1), 279–295 (2020). <https://doi.org/10.1016/j.ejor.2020.01.058>
5. S. Transchel, O. Hansen, Supply planning and inventory control of perishable products under Lead-time uncertainty and service level constraints. *Procedia Manuf.* **39**, 1666–1672 (2019). <https://doi.org/10.1016/j.promfg.2020.01.274>
6. H. Sang, S. Takahashi, R. Gaku, Big data-driven simulation analysis for inventory management in a dynamic retail environment, in *Proceeding of the 24th International Conference on Industrial Engineering and Engineering Management 2018*, (Springer Singapore, Singapore, 2019), pp. 687–694
7. S.C. Tsai, I.-Y. Ho, Sample average approximation for a two-echelon inventory system with service-level constraints. *J. Oper. Res. Soc.* **70**(4), 675–688 (Apr. 2019). <https://doi.org/10.1080/01605682.2018.1457479>

8. J.-A. Kim, J. Jeong, Simulation Evaluation for Efficient Inventory Management Based on Demand Forecast, 2018, pp. 639–650
9. F. Persson, M. Axelsson, F. Edlund, C. Lanshed, A. Lindstrom, F. Persson, Using simulation to determine the safety stock level for intermittent demand, in *2017 Winter Simulation Conference (WSC)*, 2017, pp. 3768–3779, doi: <https://doi.org/10.1109/WSC.2017.8248089>
10. M. Agumas, J. Jayaprakash, M. Teshome, Simulation study of inventory performance improvement in consumer products trade business unit using system dynamic approach, 2019, pp. 401–409
11. S.L. Takeda Berger, G.L. Tortorella, E.M. Frazzon, Simulation-based analysis of inventory strategies in lean supply chains. *IFAC-PapersOnLine* **51**(11), 1453–1458 (2018). <https://doi.org/10.1016/j.ifacol.2018.08.310>
12. A. Nagle, S. Fisher, S. Frazier, S. McComb, Streamlining a simulation center’s inventory management. *Clin. Simul. Nurs.* **18**, 1–5 (May 2018). <https://doi.org/10.1016/j.ecns.2018.01.001>
13. M. Arani, S. Abdolmaleki, X. Liu, Scenario-based Simulation Approach for an Integrated Inventory Blood Supply Chain System, *unpublished*
14. E. Eraslan, Y.T. Iç, An Improved Decision Support System for ABC Inventory Classification. *Evol. Syst.* (2019). <https://doi.org/10.1007/s12530-019-09276-7>
15. N.P. Ayu Nariswari, D. Bamford, B. Dehe, Testing an AHP model for aircraft spare parts. *Prod. Plan. Control* **30**(4), 329–344 (2019). <https://doi.org/10.1080/09537287.2018.1555341>
16. M. Ng, W.K. Talley, Chassis inventory management at U.S. container ports: Modelling and case study. *Int. J. Prod. Res.* **55**(18), 5394–5404 (2017). <https://doi.org/10.1080/00207543.2017.1315193>
17. UMMBC Series, ScottMadden. [Online]. Available: <https://www.scottmadden.com/insight/umm-bc-series/>. Accessed 9 Mar-2020
18. T.L. Saaty, What is the analytic hierarchy process? in *Mathematical Models for Decision Support*, (Springer Berlin Heidelberg, Berlin, Heidelberg, 1988), pp. 109–121
19. M. D. Rossetti, *Simulation Modeling and Arena*. 2015
20. Amarjeet, J.K. Chhabra, FP-ABC: Fuzzy-Pareto dominance driven artificial bee colony algorithm for many-objective software module clustering. *Comput. Lang. Syst. Struct.* **51**, 1–21 (2018). <https://doi.org/10.1016/j.cl.2017.08.001>
21. M. Momeni Tabar, N. Akar, D. Zaghi, H.R. Feili, M. Ghaderi, Fuzzy mathematical modeling of distribution network through location allocation model in a three-level supply chain design. *J. Math. Comput. Sci.* **9**(3), 165–174 (2014). <https://doi.org/10.22436/jmcs.09.03.02>
22. P. Pramanik, M.K. Maiti, An inventory model for deteriorating items with inflation induced variable demand under two level partial trade credit: A hybrid ABC-GA approach. *Eng. Appl. Artif. Intell.* **85**, 194–207 (2019). <https://doi.org/10.1016/j.engappai.2019.06.013>

Sustainability, Big Data, and Local Community: A Simulation Case Study of a Growing Higher Education Institution



Anatoly Kurkovsky

1 Introduction

Usually, the main idea of sustainable development (SD) methodology/paradigm is considered as a reasonable balance among at least three (in many cases four) structured components: economic, natural environmental protection, social equity [1], and “constituent” structure. The “constituent” component provides more flexibility to the SD paradigm. Today SD paradigm is applied to relatively small objects or processes as communities, cities, businesses (organizations or institutions), and also technologies [2].

Higher education institutions (HEI) are core elements in the SD paradigm. They prepare the future decision-makers of the local, national, or international societies [1]. We understand the SD paradigm application as a computational problem to analyze the numerical values of institutional goal dynamics and the possibility to reach their reasonable balance within the amount of the available resources.

By definition, HEI SD computational problems have a high complexity level. Not all researchers or researcher teams can create an HEI sustainability model from ground zero. Therefore, attempts to reduce the complexity of potential SD models without missing needed details along with creating approaches to increase the reusability of the already created models are very important.

HEI generates/collects an enormous amount of various big data related to students, professors, administrators, enrolments and retention rates, student’s success, daily operations, etc. semester by semester. Just accumulation of such big data does not automatically provide benefits to increase HEI effectiveness that potentially can be obtained within SD paradigm. However, according to B. Daniel: “whereas big

A. Kurkovsky (✉)

University System of Georgia, Georgia Gwinnett College, Lawrenceville, GA, USA

e-mail: akurkovsky@ggc.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_62

859

data is beginning to be utilized for decision making in higher education as well, practical applications in higher education instruction remain rare” [3].

There are two primary goals of this paper: (a) to present our approach that potentially can reduce the mentioned problems of SD implementation in HEIs area and adaptations of some available big data and (b) to demonstrate how our approach works within a simulation case study for analyzing a transportation system’s dynamics that is understood as a part of SD of the US young fast-growing HEI.

This paper is organized as follows. A brief review of the models related to HEI sustainability can be found in Sect. 2. A brief analysis of the case study subject domain as a conceptual system, big data, and cyberinfrastructure is introduced in Sect. 3. Section 4 describes the local community and the transportation system outside and inside of the HEI’s campus. Section 5 introduces a preliminary formalization of HEI sustainability phenomena. Section 6 describes the scope of the simulation case study and the simulation assumptions. Section 7 includes descriptions of the simulation case study where it was demonstrated how simulation could be used for the analysis of HEI sustainability, particularly in the area of the transportation systems. Section 8 provides a brief descriptions of other simulation case studies that used our simulation approach and have been already created to support the HEI sustainability.

2 Some Simulation Models Created to Analyze Higher Education Sustainability

As a rule, a set of specially created simulation models are applied to analyze various aspects of HEI sustainability. There are many publications related to particular aspects of sustainability and challenges associated with universities and colleges around the world. It is essential to address the criteria or indicators that are possible to use to assure the quality of the educational process [10] and to establish the educational system evaluation procedures [11]. What are the quality standards and quality models? It is a crucial question for higher education [12]. Modeling of various aspects of education is a conventional approach to formalize such processes. Many examples could be found in the monograph by Koper and Tattersall, dedicated to studying design and modeling [13]. Formalization and modeling of the student population is an essential element in our work. An interesting approach is to use fuzzy logic for student classification which was proposed by Nykänen [14].

Many models of transportation dynamics can be found in the Transportation Research Board [15] and the US Department of Transportation [16] websites. Some formal approaches and models related to the analysis of the parking lot dynamics can be found in relatively recent publications, Bustillos et al. [17], Batabyal and Nijkamp [18], and Brown [19], which included the examples of higher education campuses, as well as simulation analysis of the parking lots.

Modeling of various aspects of sustainable development (and simulation also) are widely used tools in this area today. There are many models with different levels of complexity that can be used to solve the existing problem of sustainable development partially. Among these models are as follows [9, 10]: (a) the TARGETS model (tool to assess regional and global environmental and health targets for sustainability) which is designed to study the issues surrounding global change and sustainable development; (b) the Asian Integrated Model (AIM), a model developed at the Japanese National Institute for Environmental Studies to study the impacts of mitigation and adaptation scenarios on the Asia-Pacific region; and (c) the CETA, a simple model developed by the Electric Power Research Institute to explore optimal combinations of abatement and adaptation policies. Also, some interested sustainability models were published by authors Kanegae [20], The Pangaea, gaming simulation model for sustainable regional development, and Spangenberg et al. [21], The Sustainability Model, European and German approaches. Detailed analysis of the later models can be found in Kurkovsky's paper [29].

However, publications mentioned above are related only to some partial formalization and modeling that could be potentially incorporated into HEI sustainable development. The publications did not describe a simulation approach that can be used as an umbrella to simulate various aspects of HEI sustainable development on a united methodological base.

3 The Case Study Subject Domain: Conceptual System, Big Data, and Cyberinfrastructure

From a systematic/conceptual point of view, SD methodology/paradigm can be considered as a balance of economic growth, environmental protection, social equity, and a constituent/architecture/structure of a particular subject domain [1, 2, 4, 5]. Therefore, within this case study, we apply the SD methodology for the Georgia Gwinnett College (GGC), a unit of the University System of Georgia, USA; it is mandatory to identify some details of the subject domain.

What is logical modularity within the GGC subject domain of our case study and future simulation models? In this case, a set of potential sustainable development solutions will be related to some structured elements/subsystems inside and outside of GGC.

Inside the GGC HEI, there are elements/subsystems as faculty members, students, learning technology, college facilities, college administrative structure, and college economy.

Outside the GGC HEI, there are elements/subsystems as market requests, government requests, environmental requests, local community, etc. (Fig. 1).

Each inside and outside elements/subsystems shown in Fig. 1 is associated with much available data that have various time scales and different granularities, and

Fig. 1 A logical border of GGC conceptual system for sustainable development analysis

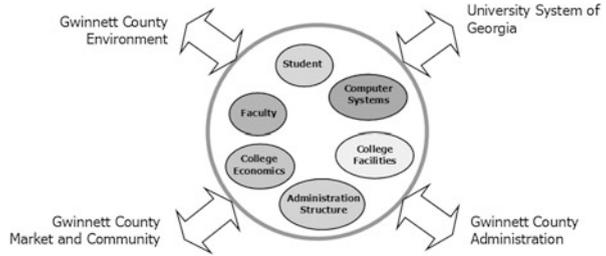


Table 1 Growing dynamics of the GGC students

	GGC year					
	2006	2007	2008	2009	2010	2011
Students	118	788	1563	2947	5380	7742
	GGC year					
	2012	2013	2014	2015	2016	
Students	9397	9719	10,828	11,347	12,052	

many of them are related with each other and have consistency problems. But at the same time, amount/volume of these data taken together is, as a rule, very big, and, from today’s methodological point of view, such data may be considered as big data.

Traditionally, big data is associated with three “v” words, which refer to (1) volume, the increasing size of data; (2) velocity, the increasing rate at which it is produced and analyzed; and (3) variety, increasing range of sources, formats, and representations [6–8]. Some authors added two extra “v” words to describe big data. According to [8], it can be (4) “veracity - to encompass the widely differing qualities of data sources, with significant differences in the coverage, accuracy, and timeliness of data” as well as (5) value, to describe how the data are useful and valuable in the organization for some purposes [3].

To increase existed veracity of the HEI data sources, we suggest existing specialized computer monitoring systems within the HEI cyberinfrastructure. These monitoring systems provide specific generalized assessments of the HEI. Results of the monitoring and retrospective information related to the HEI activities are accumulated in appropriate databases.

4 Local Community and the Higher Education Institution

The GGC is one of the youngest and fast-growing 4-year HEIs in the USA which was established in the year 2006. Growing dynamics of the GGC start from about 100 students and today reached about 13,000 students (Table 1) [23].

It is clear that such fast growth of GGC generates many serious challenges for the college facility capacity, the number of professors with various specializations

Fig. 2 Location of the Georgia Gwinnett College



needed to teach 13 thousand students, educational computer systems, and cyberinfrastructure in the campus and many transportation problems.

Most GGC students belong to the local community and live in Gwinnett County of Georgia, which is one of the most prominent suburbs of the city of Atlanta area. This creates an extra burden to the transportation infrastructure outside of the campus which needs to have a sufficient number of available parking spaces provided to the students, professors, and staff.

Therefore, at our simulation case study, we will concentrate on the transportation relationships of the local Gwinnett County community and parking lot dynamics in the GGC campus.

The GGC campus is located on the corner of the University Parkway (US 316) and the Collins Hill Road (Fig. 2).

To understand the numerical framework of the GGC sustainability transportation constraints, we need to analyze the transportation system dynamics for the intersection of SR 316 and Collins Hill Road. The Gwinnett County Department of Transportation (DOT), GA, provided us with statistical data about the traffic dynamics at the intersection. A sample of the data for the evening hours is shown below (Fig. 3).

As shown in Fig. 3, traffic jams were generated during the peak hours for several directions: eastbound left (EBL), eastbound traffic (EBT), eastbound right (EBR), northbound left (NBL), northbound traffic (NBT), and northbound right (NBR).

5 Sustainability Phenomena Preliminary Formalization

As previously mentioned, SD includes four structured components: economic (EC), social (SL), environmental (EN), and constituent (CT). Each of the structured components can be described with a set of parameters/variables/vectors. Therefore,

1: SR 316 & Collins Hill Road Baseline - Default



Lane Group	EBL	EBT	EBR	WBL	WBT	WBR	NBL	NBT	NBR	SBL	SBT	SBR
Volume (vph)	432	2676	104	43	1361	101	308	381	72	138	169	40
Confl. Peds. (#/hr)												
Confl. Bikes (#/hr)												
Peak Hour Factor	0.97	0.97	0.97	0.89	0.89	0.89	0.92	0.92	0.92	0.76	0.76	0.76
Growth Factor	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Heavy Vehicles (%)	2%	2%	2%	2%	2%	2%	2%	2%	2%	2%	2%	2%
Bus Blockages (#/hr)	0	0	0	0	0	0	0	0	0	0	0	0
Parking (#/hr)												
Mid-Block Traffic (%)		0%			0%			0%			0%	
Adj. Flow (vph)	445	2759	107	48	1529	113	335	414	78	182	222	53
Shared Lane Traffic (%)												
Lane Group Flow (vph)	445	2759	107	48	1529	113	335	414	78	182	222	53
Intersection Summary												

Fig. 3 Some statistical data related to the GGC transportation system dynamics (evening hours). Source: Gwinnett County DOT, GA

generally, from a computational point of view, SD can be presented as a rational sustainable balance (RSB) among these four sets of related vectors:

$$RSB = (\{SL\}, \{EC\}, \{EN\}, \{CT\}) .$$

Each vector can include some (n) variables that describe a particular aspect of the SD structured components:

$$SL = \{SL1, SL2, \dots, SLn\}$$

$$EC = \{EC1, EC2, \dots, ECn\}$$

$$EN = \{EN1, EN2, \dots, ENn\}$$

$$CT = \{CT1, CT2, \dots, CTn\}$$

At this pre-formalization level, we need to clarify what we considered as a system for our future SD analysis and simulation. The term “system” is widely used with various meanings in some scientific disciplines or many case studies. According to Klir and Elias suggestion, “system (S) stands for a set of some things (A) and relations (R) among the things” [22]. Therefore a system can be described as

$$S = (A, R) .$$

By using this description, Klir and Elias propose to divide the systems into two classes: (a) first, systems based on certain kinds of things and, (b) second, systems based on certain kinds of relations.

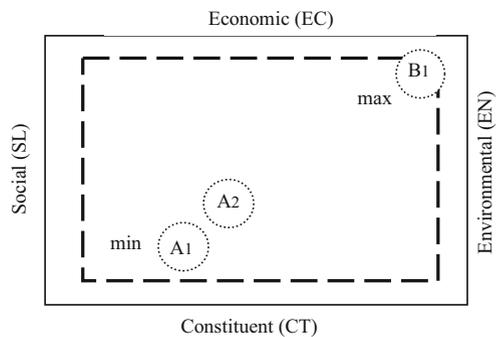
The first class describes a set of traditional systems classification by science or technology to investigate certain kinds of things: environmental, social, economic, or constituent. Usually, such systems require some experimental procedures for data acquisition and can be viewed as “accentually experimentally based.” The second class describes a set of systems characterized by specific kinds of relations and therefore required “intensive data processing rather than data acquisition” [22].

For SD analysis and simulation, we consider to use the second class of particular systems (SD structured components): {SL}, {EC}, {EN}, and {CT}. We used this class of the systems because by SD definition, it is required to find a rational balance among them (specific relations and their dynamics among the systems parameters/variables).

A magnified view of HEI SD metrics of EC-, SL-, EN-, and CT-structured components (abstract systems) is shown in Fig. 4. The left bottom corner on a rectangle of Fig. 4 represents an area with a minimum level of relations among the four SD structured components. The right upper corner of the rectangle represents an area with a maximum level of relations among the four structured components.

Figure 4 demonstrates two small dotted circles A1 and A2 that represented a set of conducted HEI sustainable development simulation case studies. A set of previously conducted simulation case studies (dotted circle A1) is briefly described in Sect. 7 of this paper. The dotted circle A2 represents a set of the simulation case studies briefly mentioned in Sect. 8 of the paper. A small dashed circle (B1) represents a whole (max) level of all CD components. It will be a place of a set of future simulation models with a maximum level of relations among the EC-, CT-, SL-, and EN-HEI sustainable development-structured components.

Fig. 4 HEI sustainability metrics and their association with a set of simulation case studies



6 Scope of the Simulation Case Study and the Simulation Assumptions

Our simulation model should help us to analyze the transportation system dynamics outside and inside of the college campus. We formulated several questions to analyze how the GGC can impact to the local community transportation system dynamics outside of the college campus and what kind of balanced parking lot capacity the college campus should have to minimize potential delays of the arriving time of the students and professors.

1. What are the numerical values of the GGC sustainable growth constraints due to the current transportation system: total student enrolment and staff (number of professors, administration, and service staff) for each semester (for 3 years)?
2. What is the influence of the GGC growth parameters on the numerical value of the traffic dynamics on the section of SR 316 depending on the period of day (in the morning and in the evening) for a day-by-day range incorporated into a semester-by-semester range (for 3 years)?
3. What is the numerical value of the dynamics of crucial traffic constraints (traffic jam duration) related to the current structure of the transportation system (traffic light) and the college growth parameters depending on the period of day (in the morning and in the evening) for a day-by-day range incorporated into a semester-by-semester range (for 3 years)?
4. What kind of potential changes of current GGC parking lot system structure will be needed to accommodate about 15,000 students on campus with the appropriate number of faculty/staff, visitors, and handicapped?

Any simulation model is an abstraction of the real-world processes and/or objects. Therefore, any simulation model will be designed by using some assumptions in order to reduce the complexity of the portion of the real world to be analyzed and simulated. For our simulation model, we used the following set of assumptions:

- Duration (a step) of the simulation is one semester. The maximum of six steps are needed to simulate.
- Traffic variables have a set of values hierarchies with minutes, hours, days, and months.
- During each step, the value of the main simulation variables cannot be changed.
- Changes of variable values are possible at the beginning of the semester.
- Traffic variables (a vehicle in the traffic considered as an entity) have five stochastic parameters: (a) input distribution before the traffic light; (b) time when this entity was generated (arriving time in the simulation system); (c) probability to turn on the traffic light to the right or left or to go straight; (d) probability to enter the college campus; and (e) waiting time in a queue before the traffic light (initially considered with a null value).

- This simulation model can be used independently or can be considered as a low-level computational component of a simulation model hierarchy.
- A set of the numerical results generated by this simulation model can be used as an input for a higher education institution sustainability simulation at the top level of the simulation model hierarchy.

7 The Simulation Case Study: Simulated Processes, Implementation of the Simulation Model, and Some Numerical Results

In order to describe how our simulation model is organized, we need to explain what kind of information flows will be processed within the model and what kind of statistical data we have used. For our simulation, we used the following information:

1. Statistical data (vehicles amount/time unit: minutes, hours, days, months) related to the traffic light.
2. Statistical data related to the traffic light for four directions from and to Atlanta (SR 316: westbound-eastbound) and from and to Lawrenceville (Collins Hill Road: southbound-northbound) before the traffic light.
3. Description of the traffic light changes procedure (a duration time for red and green, including arrows and its dynamics minute by minute, hour by hour, for a whole day). For some cases, if it varies for different days of the week, we used a set of histograms for each day of the week or a month.
4. A range of allowed delay value to be in a GGC class late but do not miss the class.
5. Statistical information (customers: students, faculty/staff, handicapped, and visitors) related to the number of customers arriving on campus.
6. Maximum parking space capacity of all parking lots located on campus.
7. A set of forecasted values for the GGC enrolment semester by semester during the next six semesters (for 3 years).

Logical schema of our simulation model is briefly described below (Fig. 5).

7.1 *Students/Customer Arriving Processes*

A new vehicle generator produces a sequence of entities (cars with a person constituted the transportation traffic) with five incorporated stochastic parameters as described earlier for northbound, southbound, westbound, and eastbound accordingly. Four vehicle generators are shown at the leftmost lower portion of the section “a” of the Fig. 5. For simplicity, only two vehicle parameters (a probability to turn

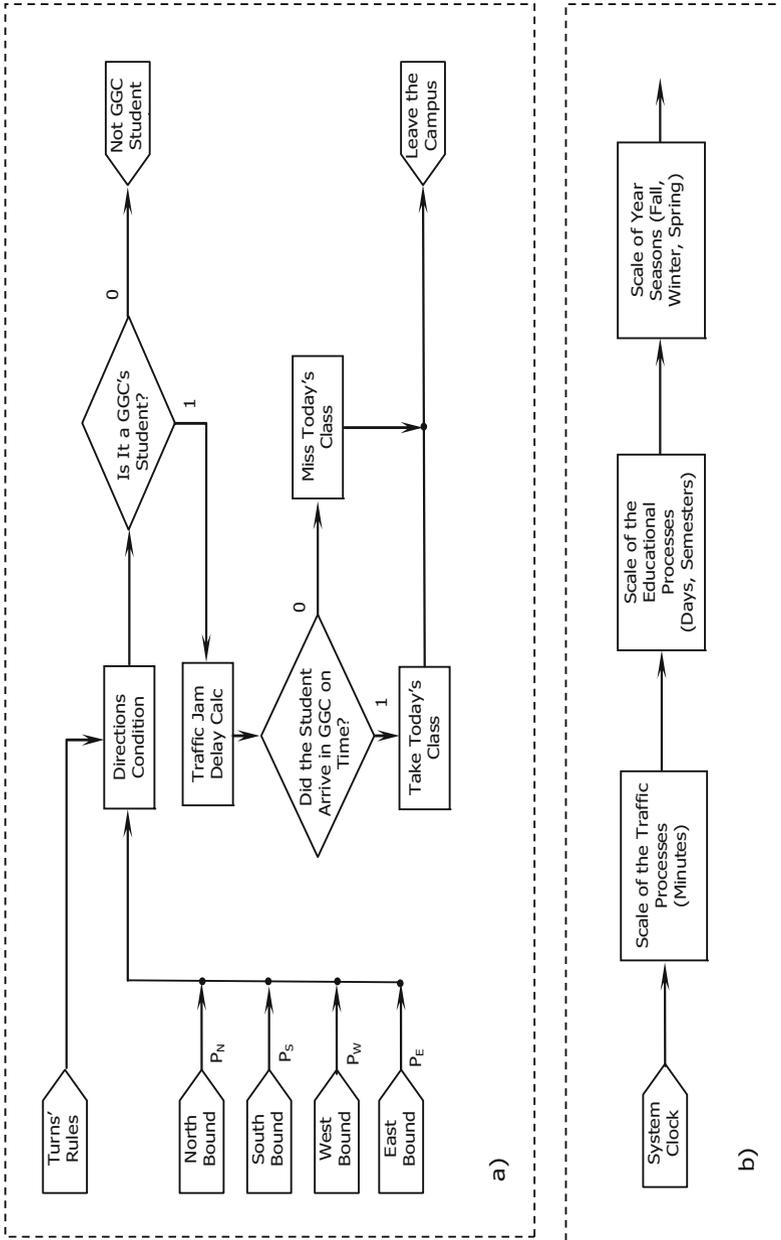


Fig. 5 Logical schema of the simulated processes: (a) the students arriving process and (b) the time scales for the transportation system dynamics analysis

on the traffic light to right or left or to go straight and a probability to enter the college campus) are shown in the Fig. 5 as PN, PS, PW, and PE.

A generator of the traffic light change procedure produces a sequence of entities associated with “green,” “red,” “yellow,” various “arrows,” and its dynamics (appearance duration) minute by minute. This generator is shown as Turn’s Rules block at the leftmost upper portion of the section “a” of the Fig. 5.

Direction Conditions block allows us to associate the current system simulation time with two entities: current vehicle and current traffic light condition. From an abstraction point of view (queuing system theory terminology), here we have a queue that generates potential traffic jams.

After passing the traffic light, each entity vehicle should be checked for GGC membership by the block titled “Is it a GGC student?”

Not GGC Student block will terminate the entity (in case of false) by removing it from the simulation model.

Traffic Jam Delay Calculation block will process the entity (in case of true) to obtain a value of traffic jam delay for this entity. After passing the Traffic Jam Delay Calculation block we need to check the entity accumulated delay with the known range of allowed delay value to be in a GGC class late but do not miss the class. The block titled “Did the Student Arrive on Time?” provided this procedure.

Missed Today’s Class block will accumulate the number of the students who missed GGC classes (in case of false) because of the traffic jam.

Take Today’s Class block will accumulate the number of the students who arrived to GGC on time (in case of true).

Leave the Campus block will terminate the entity by removing it from the simulation model.

7.2 Time Scales Process

The simulation system time clock will generate various time scales needed to simulate the GGC transportation dynamics, section “b” of the Fig. 5. Several different scales will be applied in the model: minutes, hours, days, months, semesters, seasons of the year, and years. Time scale process provides a synchronization for all entities generated within the simulation model.

7.3 Implementation of the Simulation Model Preliminary Version

There are different ways to implement the described above processes within several professional simulation programming environments. Brief information about professional simulation software can be found at many publications [24, 25]. We used

Arena professional software to implement our model. Arena was chosen because it is not only a simulation software package but it also includes professional simulation language SIMAN. Arena provides a user-friendly interface to organize simulation modeling with various scenarios.

7.4 Case Study Simulation: Some Results

A sample of the simulation results of the described above case study is shown in Fig. 6.

The Fig. 6 clearly demonstrates that during the time between 10 am and 8 pm, all total parking spaces were occupied by GGC customers at the level of 80–98%. As a result, for this period, only 75–80% GGC customers successfully arrived to campus. Remember, for this simulation scenario, we used 15 minutes limit for a successful customer. To be successful, the customer could spend less than 15 minutes (a) to find free space on a parking lot and (b) to walk to a target campus building.

The above simulation numerical results are compatible with the actual parking lot observations on campus, and they provide us with a preliminary confirmation that we used a correct formalization approach and made reasonable assumptions to create the simulation model.

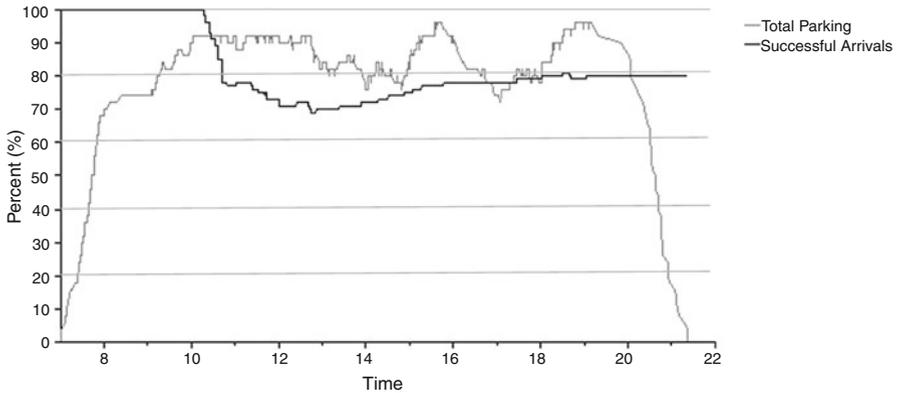


Fig. 6 Sample of a dynamics of the parking availability and the allocated parking spaces on the campus

8 Other Simulation Case Studies to Support the Higher Education Institution Sustainability

Because of a sustainable development simulation complexity for whole GGC HEI, we decomposed it into a set of particular subject domains. For the identified subject domains, we conducted many GGC HEI simulation case studies. They are shown in Fig. 1, as a dotted circle A1 and A2, which represented relatively stable EC-CT relations with a small level of SL and EN relations within the GGC HEI. Some of the previously created simulation case studies include:

- A simulation model for the educational process management and to support college sustainability [28]
- A simulation model to analyze the potential damage of an Intranet Worm Outbreak within the local college network [27]
- A simulation model to analyze the curriculum quality decision-making procedures and the associated risk assessments to support the college sustainability [26]

9 Conclusion

We proposed a simulation approach that can be used as an “umbrella” for various aspects of HEI sustainable development and big data incorporations and to conduct an HEI impact analysis to some outer subsystems.

We illustrated how the proposed simulation approach works within a simulation case study where sustainability, big data, and local community are combined together to analyze the transportation system dynamics associated with a very fast-growing 4-year Georgia Gwinnett College, a unit of the University System of Georgia, USA.

References

1. Higher Education Sustainability Initiative (HESI), a Partnership of UN Sustainable Development Organizations. <https://sustainabledevelopment.un.org/sdinaction/hesi>. Accessed 5 Mar 2020
2. S. Sikdar, Sustainable development and sustainability metrics. *AIChE J.* **49**(8), 1928–1932 (2003)
3. B. Daniel, Big data in higher education: The big picture, Chapter 3, in *Big Data and Learning Analytics in Higher Education: Current Theory and Practice*, ed. by B. K. Daniel, (Springer International Publishing, Cham, 2017)
4. UN Initiatives to Unleash Big Data for Sustainable Development. <http://sdg.iisd.org/news/un-launches-initiatives-to-unleash-big-data-for-sustainable-development/>. Accessed 10 Apr 2020
5. N. Younis, Big data and sustainability of higher education, Chapter 4, in *Global Approaches to Sustainability Through Learning and Education*, (IGI Global, Hershey, 2020)

6. D. Laney, 3D Data Management: Controlling Data Volume, Velocity, and Variety, Application Delivery Strategies (META Group), 6 Feb 2001
7. X. Dong, D. Srivasta, Big data integration, In *ICDE Conference 2013*, Brisbane, Australia, 8–11 Apr 2013
8. C. Dede, A. Ho, P. Mitros, Big data analysis in higher education: Promises and pitfalls. *EDUCAUSE Rev* **51**(5), 18–32 (2016)
9. P. Corcoran, A. Wals (eds.), *Higher Education and the Challenge of Sustainability: Problematics, Promise, and Practice* (Kluwer Academic Publishers, Dordrecht, 2004)
10. R.A. Ellis, A. Calvo, Minimum indicators to assure quality of LMS-supported blended learning. *Educ. Technol. Soc.* **10**(2), 60–70 (2007)
11. R. Lanzilotti, C. Ardito, M.F. Costabile, A. De Angeli, eLSE methodology: A systematic approach to the e-learning systems evaluation. *Educ. Technol. Soc.* **9**(4), 42–53 (2006)
12. J.M. Pawlowski, The quality adaptation model: Adaptation and adoption of the quality standard ISO/IEC 19796-1 for learning, education, and training. *Educ. Technol. Soc.* **10**(2), 3–16 (2007)
13. R. Koper, C. Tattersall, *Learning Design. A Handbook on Modelling and Delivering Networked Education and Training* (Springer, Berlin, Heidelberg, 2005)
14. O. Nykänen, Inducing fuzzy models for student classification. *Educ. Technol. Soc.* **9**(2), 223–234 (2006)
15. Transportation Research Board (TRB). Retrieved 1 Oct 2010 from: <http://www.trb.org/Main/Home.aspx> Accessed May 04, 2020
16. US Department of Transportation - Federal Highway Administration (FHWA). Publications. <http://ops.fhwa.dot.gov/publications/publications.htm>. Accessed 7 May 2020
17. B. Bustillos, J. Shelton, Y.-C. Chiu, Urban university campus transportation and parking planning through a dynamic traffic simulation and assignment approach. *Transport. Plan. Technol.* **34**(2), 177–197 (2011)
18. A. Batabyal, P. Nijkamp, A probabilistic analysis of two university parking issues. *Ann. Region. Sci.* **44**(1), 111–120 (2010)
19. J. Brown, D.B. Hess, D. Shoup, Fare-free public transit at universities an evaluation. *J. Plan. Educ. Res.* **23**(1), 69–82 (2003)
20. H. Kanegae, *Pangaea – Gaming Simulation Exercise for Sustainable Regional Development*, UNCRD Training Material Series, No. 2 (United Nations Centre for Regional Development, Nagoya, 1998)
21. J.H. Spangenberg, I. Omann, A. Bockermann, B. Meyer, Modelling sustainability – European and German approaches, in *Integrative Systems Approaches to Natural and Social Dynamics*, (Springer Verlag, Berlin/New York, 2001), pp. 481–503
22. G. Klir, D. Elias, *Architecture of Systems Problem Solving*, 2nd edn. (Kluwer Academic/Plenum Publishers, Boston, 2003)
23. GGC 2019. Georgia Gwinnett College, a unit of the University System of Georgia. GGC at a Glance. <http://www.ggc.edu/about-ggc/at-a-glance/index.html>. Accessed 10 Jan 2020
24. D. Kelton, R. Sadowski, N. Zupick, *Simulation with Arena*, 6th edn. (McGraw-Hill Education, 2015)
25. A. Law, *Simulation Modeling and Analysis*, 5th edn. (McGraw-Hill, New York, 2015)
26. A. Kurkovsky, A simulation approach to the decision-making structures analysis to support curriculum quality for higher education sustainability, in *Proceedings of “The 2016 Summer Simulation International Conference (SummerSim’16)*. Montreal, Quebec, Canada, pp 409–414, 24–27 July 2016
27. A. Kurkovsky, Interdisciplinary systems and simulation studies for an innovative undergraduate program, in *Proceedings of ‘Emerging M&S Applications in Industry and Academia (EAI 2013) Symposium, SpringSim 2013’*. San Diego, CA, USA. pp 580–588, 2013
28. A. Kurkovsky, T. Mundie, Simulation of educational process management to support sustainability of an innovative higher education, in *Proceedings of the 20th International Conference on Modeling and Simulation (MS 2009)*. Banff, Alberta, Canada, pp 435–448, 2009
29. A. Kurkovsky, Simulation in analysis of sustainable development: Goals, planning, and evaluations, in *Proceedings of the International Conference on Modeling, Simulation, and Optimization (MSO’04)*. Kauai, Hawaii, USA. pp. 275–280, 2004

Vehicle Test Rig Modeling and Simulation



Sara Boyle

1 Introduction

Modeling and simulation of vehicle dynamics are widely used in commercial and military vehicle applications to test existing and future vehicle designs and identify key mobility metrics at a lower cost than physical testing. Modeling and simulation tools can also be used at an earlier point in the vehicle design life cycle when physical vehicle components are not yet available. For a high-fidelity, multi-body physics-based vehicle model, there are many vehicle components that will need specification so it is best to be able to start with a smaller set of information to ensure that the vehicle model is accurate and complete. The vehicle test rigs focus on a piece of the overall vehicle model to ensure that the vehicle dynamics are working properly and that the model is complete. The vehicle subsystem test rigs that will be the focus of this paper are suspension, tire, and track test rigs.

When initially designing modeling and simulation tools, it is important to perform verification and validation of these tools by comparing it to the real-world test data with trusted models. These test rigs can also be used to verify that the multi-body physics-based modeling is performing in an expected way that reflects physical testing and the results of other commercial modeling tools.

S. Boyle (✉)

Ground Vehicle Systems Center (GVSC), Combat Capabilities Development Command (CCDC),
Warren, MI, USA

e-mail: sara.e.boyle.civ@mail.mil

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_63

873

2 Vehicle Test Rigs

2.1 Physical Test Rig Systems

Physical systems for testing vehicle subsystems include individual tire test rigs and hydraulic post-automotive test systems. Tire test rigs have the ability to attach a single tire and wheel assembly to a rig that can then perform testing to characterize the tire performance on- and off-road. Hydraulic post-automotive test systems have a hydraulic actuator for each tire of the vehicle that can be independently manipulated to analyze the vehicle performance. They can be used to represent the characterization of a road and test the vehicle suspension capabilities.

Four poster automotive test systems are common for evaluating commercial vehicles; however for military vehicles, many of the vehicles have more than four wheels so they require larger test systems to test all of the wheel and suspension components at the same time. This specialized equipment can be very costly to run and requires the use of a full vehicle.

2.2 Modeling and Simulation of Test Rigs

Modeling and simulation of the vehicle test rigs can offer analysts many opportunities to save resources. Modeling and simulation can offer opportunities to test vehicle designs that have not been yet been physically implemented to ensure that the design performs as expected. There is also a shorter turnaround time on creating a model and testing it virtually rather than go through the cost and time of creating a physical system and testing it.

The virtual tire test rig closely mirrors the physical test rig system available and models a single tire-and-wheel assembly. The suspension and track test rigs are similar to the hydraulic post-automotive test rigs, but they only focus on one suspension or track assembly and do not require the entire vehicle model be specified. The virtual hydraulic post-implementations can be used to evaluate the vehicle design using real-world road profile data as well.

3 Method

A description of the method used for the test rig implementation using Chrono, Mercury, and Ground Contact Element (GCE) is described in the next sections.

3.1 Chrono

Chrono [1] is an open source, multi-body physics-based middleware library that can be used for modeling and simulation. Chrono has a module called Chrono::Vehicle [2] that supports template-based modeling for both wheeled and tracked vehicles. The vehicle is an assembly of vehicle subsystems, and each subsystem can be described with its own JavaScript Object Notation (JSON) [3] file. A variety of subsystem templates are available for each vehicle component. For instance, there are multiple types of vehicle suspension templates available, such as double wishbone and multi-link, that give the user the ability to define information based on a pre-defined suspension topology.

The Chrono::Vehicle module also provides some vehicle test rig classes. There are tire and suspension test rigs for wheeled vehicles and a track test rig for the tracked vehicles. To use these test rig classes, the user would have to implement them using C++. This may be accessible to some users, but many of the analysts that would be performing vehicle dynamics analysis do not have the programming background to implement this themselves. The test rig implementation described in this paper uses the underlying Chrono::Vehicle test rig class structure and makes them available to users through input files rather than have the user implement their own C++ implementation of the vehicle test rigs.

3.2 Mercury

Mercury [4] is a co-simulation framework that supports using Chrono and other vehicle component modeling codebases for vehicle mobility test modeling and simulation. For the test rig implementation described in this paper, the full Mercury framework is not being used; however, many Mercury utilities are being used to parse the input file information and provide the user with meaningful output messages throughout the simulation and for data logging the output from the test rig simulation.

The test rig input file used by the test rig simulation is a JSON file format. Mercury makes use of a similar input file, and there are utilities in Mercury to access the data from this input file. These utilities are used in the test rig implementation to make the options input file more flexible for the user to create by utilizing standardization of key names and string values. The options file utilities also make it very easy for future development efforts to add more options to the input file if there are new features of the test rig simulation that the user would like to expose through the input file.

The meaningful output messages that are printed out during the simulation also utilize a Mercury utility. The information helps inform the user of the default values being used in the simulation and what selections have been used from the user input file.

The data logging for the test rigs also use a utility class from Mercury. This should make it easy for future development to adapt to add more logging options if that is desirable. To enable logging capabilities, the user must specify an output file name in the test rig input file.

3.3 Ground Contact Element

Another unique aspect of this implementation of the test rigs is that it enables the use of Ground Contact Element (GCE) [5] tire models. This will require GCE to be built separately and identified when building the test rig implementation. These tire models are designed for use with soft soil modeling.

3.4 Test Rig Class

The main test rig wrapper class was created to provide the user with only one executable. This design decision was made in an effort to simplify the user experience since all of the types of test rigs can run with the same executable and an input file. The user can specify which type of test rig to run through the input files to run simulations with the separate suspension, tire, and track test rigs.

A wrapper class was created for each test rig type available in Chrono::Vehicle so that the user can control various parameters relevant to each different test rig. The test rigs can take input files for easier user interaction with the Chrono::Vehicle test rigs and Chrono::Vehicle components which allows the user to specify information in a human readable JSON format and does not require them to modify code and recompile. There are pre-defined scenarios available as well as interactive options for the various test rigs. Each option that is available for the user to input was given a default value wherever possible to ensure that the user would not have to specify every single value if they did not want that level of control. This gives the user a lot of flexibility when they are running these test rigs and eases them into using the test rigs while also allowing them to fine-tune the control that they have with the test rigs as they become more familiar with the setup.

4 Building Test Rig Simulation

To build this test rig implementation, the first step is to obtain the Chrono, GCE, and Mercury repositories and build each of those first. Then it is important to obtain the vehicle's repository as well because that will be the directory used for the vehicle input data.

Next, obtain the test rig's repository and then perform the following steps:

```
> cd testrigs
> mkdir build
> cd build
> cmake ../
```

When the cmake window opens up, set the paths to the previously built Chrono, GCE, and Mercury repositories in addition to the directory for the vehicle repository. Then build the test rigs by running the following:

```
> make
```

5 Running Test Rig Simulation

From the built directory, run the test rig simulation with the following:

```
> ./test_rigs inputfile.json
```

The input file options available for each test rig will be discussed in further detail below.

The option “rig type” is required for the input file and should be set to “tire,” “suspension,” or “track” depending on which test rig the user is running.

The following are optional user input file options for all of the different rigs:

- Step size (default 1e-3 s)
- Output step size (default 0.01 s)
- Output file—must specify this to have an output file created

The following sections will discuss the differences between the different test rigs implemented.

5.1 Tire Test Rig

The tire test rig takes a single tire and runs it along an 8-meter course. There are a variety of user options that can be added to the JSON input file for the tire test rig. They are described in more detail below.

5.1.1 Tire Test Rig Input File

A “tire file,” “wheel file,” and “scene type” (described more below) are required input file options specific to the tire test rig. Here is an example of the JSON input file for the tire test rig:

```

{
  "rig type": "tire",
  "tire file": "path/to/tire.json",
  "wheel file": "path/to/wheel.json",
  "scene type": "user defined"
}

```

5.1.2 Optional Input File Options

The following are optional user input file options for the tire test rig:

- Normal load (default 8000 N)
- Camber angle (default 0)
- Tire step size (default 1e-4 s)

5.1.3 Tire Test Rig Scene Types

There are many different scene types for the tire test rig. Here are the scene-type options:

- Driven—sets angular speed to 10.
- Pulled—sets the longitudinal speed to 1 and angular speed to 0.
- Immobilized—sets the longitudinal speed and angular speed to 0.
- Motion—sets the longitudinal speed to 0.2, angular speed to 10, and the slip angle to a sine function.
- Slip—initializes the system with a longitudinal slip of 0.2.
- User Defined—the user can specify the longitudinal speed, angular speed, and slip angle values in the input file.

5.1.4 Tire Test Rig Tire Types

The following is a list of tire template types that can be used with the tire test rig:

- FialaTire
- RigidTire
- TMeasyTire
- Pacejka
- Pac02Tire
- GCE

5.2 *Suspension Test Rig*

The suspension test rig simulates a hydraulic post-actuator for one suspension system, including wheels, tires, and steering subsystems.

5.2.1 **Suspension Test Rig Input File**

A “left tire file,” “right tire file,” and “vehicle file” or “suspension test file” are required input file options specific to the suspension test rig. If the vehicle file is given, the user must specify an axle index. The “suspension test file” is not the same as just a Chrono::Vehicle suspension model. It is a JSON file that includes suspension, steering, and wheel subsystem input files as well as location and orientation information. The “suspension test file” has a Chrono::Vehicle template type of “SuspensionTestRig.” The tire template types supported by the suspension test rig are the same as the tire test rig. Here is an example of the JSON input file for the suspension test rig:

```
{
  "rig type": "suspension",
  "left tire file": "path/to/tire.json",
  "right tire file": "path/to/tire.json",
  "vehicle file": "path/to/vehicle.json",
  "axle index": 1
}
```

5.2.2 **Suspension Test Rig Driver**

The default for the driver is the GUI driver provided through Irrlicht, and it will be used if the “driver file” is not specified. A data driver can be used by providing the “driver file” in the input options file. The data driver should contain four columns with time, left post-displacement, right post-displacement, and steering inputs.

5.3 *Track Test Rig*

The track test rig simulates a hydraulic post-actuator for one track assembly system, including road wheel, sprocket, idler, and track pad components.

5.3.1 Track Test Rig Input File

A “track file” is the required input file option specific to the track test rig. Here is an example of the JSON input file for the track test rig:

```
{
  "rig type": "track",
  "track file": "path/to/trackassembly.json"
}
```

5.3.2 Track Test Rig Driver

The default for the driver is the GUI driver provided through Irrlicht, and that will be used if the “driver file” or “road file” is not specified in the input file.. The “driver file” works similarly to the suspension test rig, but the “driver file” columns will now be time, displacement of post 1, displacement of post 2, . . . , displacement of post n, and throttle, where there are n posts total. Otherwise, if “road file” is given, then the option “road speed” also needs to be included in the input file, and the road profile input data inside of the “road file” is assumed to contain (x,z) pairs, with x locations in increasing order.

References

1. <https://github.com/projectchrono/chrono>
2. R. Serban, M. Taylor, D. Negrut, A. Tasora, Chrono: Vehicle Template-Based Ground Vehicle Modeling and Simulation. Technical Report TR-2016-10, Simulation Based Engineering Lab, University of Wisconsin, Madison (2016)
3. ECMA, The JSON data interchange format. Technical Report ECMA-404, ECMA International (2013)
4. C. Goodin, J. Mange, S. Pace, T. Skorupa, D. Kedziorek, J. Priddy, L. Lynch, Simulating the mobility of wheeled ground vehicles with mercury. Technical report, SAE International Journal of Commercial Vehicles 10 (2017)
5. D.C. Creighton, G.B. McKinley, R.A. Jones, R.B. Ahlvin, Terrain Mechanics and Modeling Research Program: Enhanced Vehicle Dynamics Module. DTIC Document (2009)

Modelling and Simulation of MEMS Gyroscope with Coventor MEMS+ and MATLAB/Simulink Software



Jacek Nazdrowicz, Adam Stawinski, and Andrzej Napieralski

1 Introduction

MEMS and ASIC are often designed by two or more teams which use own engineering software. Cadence software is often used to design ASIC. Frequently MEMS designers use their own CAD software for modeling and simulation. Use of other software for modeling and simulation MEMS causes additional actions to be done. In case of many model modifications and simulations, repeated model update in Cadence is not time effective. Authors decided to use Coventor MEMS+ software for MEMS designing which cooperates with Cadence software and has additional interface to MATLAB/Simulink. This enables programming and creation of unified application for MEMS modeling and simulations.

The heterogeneous environment was created by authors for MEMS and ASIC teams (Fig. 1). MEMS designer uses Coventor MEMS+ for modeling structure and performing simple simulations; MATLAB is used for scripting/programming simulation actions and visualizing results and Simulink for performing transient simulations. All updates applied to structure can be immediately updated in MATLAB/Simulink/Cadence software, because all of them use the same .3dsch structure file. This environment is used for modeling and simulation and to create final design of the MEMS Gyroscope structure. GUIDE (from MATLAB/Simulink) environment was used by authors to create user-friendly application for input parameters, running different kind of simulations and displaying results (whole code is included here) (Fig. 2). Part of simulations is performed with Simulink model presented in Fig. 3 (with use specific MEMS+ block).

J. Nazdrowicz (✉) · A. Stawinski · A. Napieralski
Department of Microelectronics and Computer Science, Lodz University of Technology, Lodz,
Poland
e-mail: jnazdrowicz@dmcs.pl; astawinski@dmcs.pl; napie@dmcs.pl

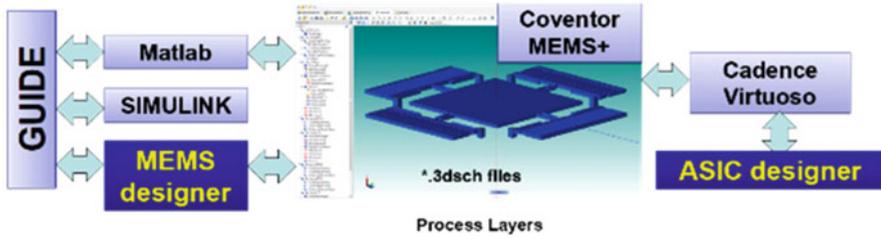


Fig. 1 Modeling and simulation heterogeneous environment

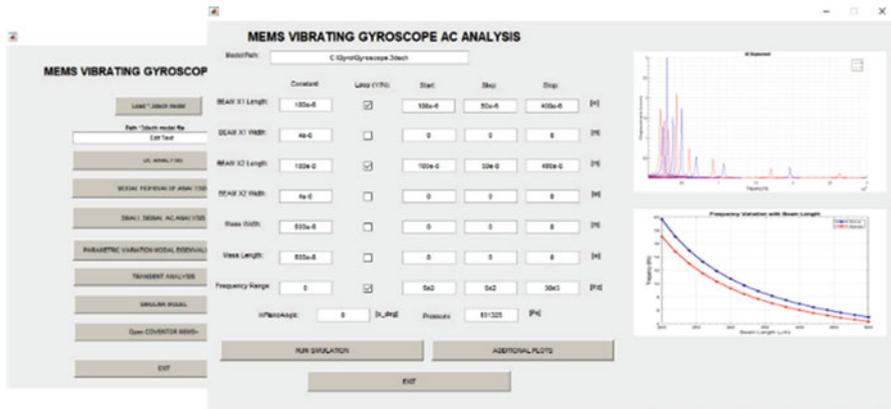


Fig. 2 Simulation application in GUIDE

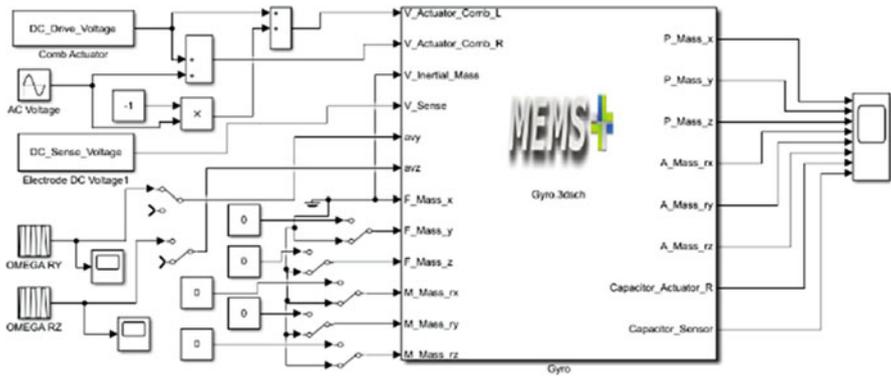


Fig. 3 Simulink model of MEMS Gyroscope with use MEMS+ block

2 Model of Gyroscope and Simulations Results

Simple model of MEMS Gyroscope is presented in Fig. 4. It vibrates along x direction and measures rotation around y axis (Coriolis force appears and acts along z direction). It consists of four beams anchored at the ends and have so-called box beams included. Box beam springs application ensures better stability for motion in z direction in presence rotation. Box beam spring also takes much less space than traditional serpentine spring.

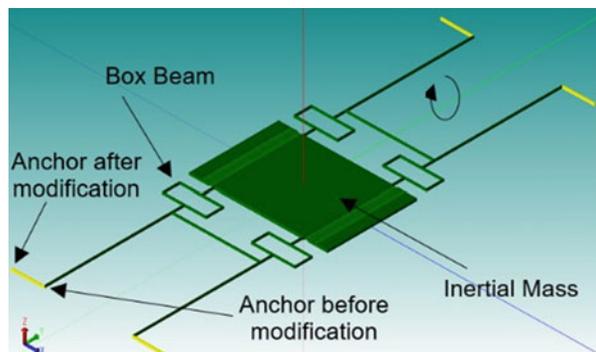
Results of this gyroscope simulation performed in created environment will be shown here.

One of the main problems in MEMS Gyroscope is to obtain signal in sense direction as high as possible. We know that when external frequency of vibration is equal to eigenfrequency – amplification is the strongest. Our research including simulation process is focused also on modal analysis. In Fig. 5, there are graphical results of such analysis for given DC simulation presented. For operational objectives, only first four nodes should be considered and their respective eigenfrequencies. Because modal analysis shows that the best for x direction is mode 1 and for z direction mode 2 – these ones were used for further simulations. Next simulations were performed for different geometrical dimensions of crucial parts of the structure (inertial mass, beam boxes, and beams). In Fig. 6, there are example results for different inertial mass dimensions (300, 400 μm to 500 μm – marked solid, dashed with dots, and dashed lines, respectively).

AC response analysis shows shift maximum displacement (for each direction) to lower frequencies. Moreover, what is very important, for frequencies scope where displacement for X and Z reaches maximum, Y displacement is the lowest (Y displacement influences on stability actuation along X direction). Additionally, change in inertial mass dimensions does not influence meaningfully on amplitude at eigenfrequency.

In Fig. 7, there are transfer function (from force to displacement) results presented. One can see that the maximum amplitude is around 10 kHz which was confirmed in modal analysis (first mode).

Fig. 4 Gyroscope model used in simulations



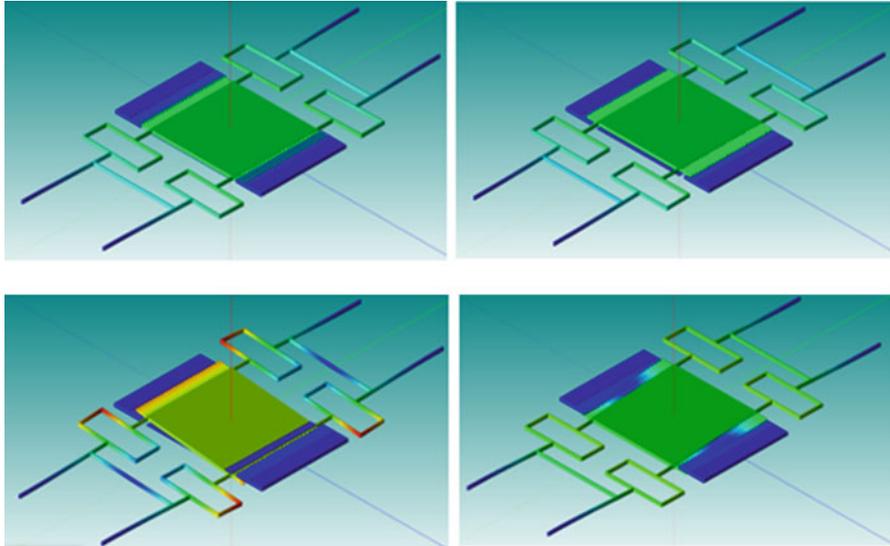


Fig. 5 Modal analysis results (particular modes are presented)

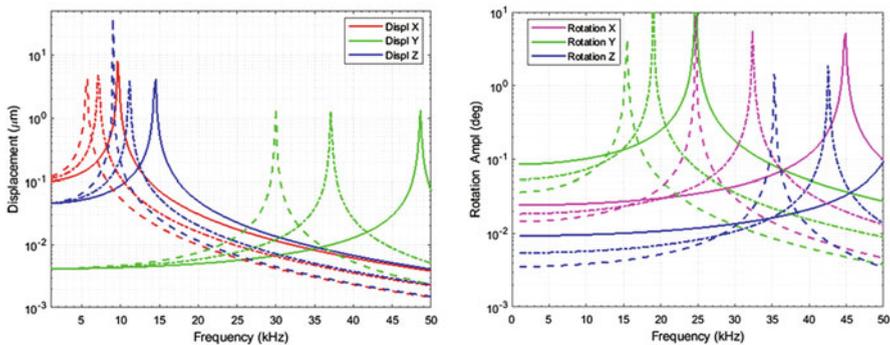


Fig. 6 AC results for different inertial mass dimensions

In the magnitude plot, at low frequencies, the value is equal to $1/(m\omega_0^2)$ or $1/k$. When the resonance frequency is approached, the magnitude of the transfer function begins increasing. At the resonance frequency, value of transfer function H is Q (quality factor) times larger and is equal to $Q/(m\omega_0^2) = 1/(D\omega_0)$ or Q/k . At high frequencies, the magnitude rolls off dramatically. In the phase plot, the phase starts at 0° and reaches a phase shift of 90° at the resonance frequency and finally 180° at high frequencies.

Results of transient analysis for duration time equals 0.05 s showed that there are many unexpected fluctuations along z direction (sense) which degrades rotation velocity measurement. We decided to add short arm perpendicular to long one

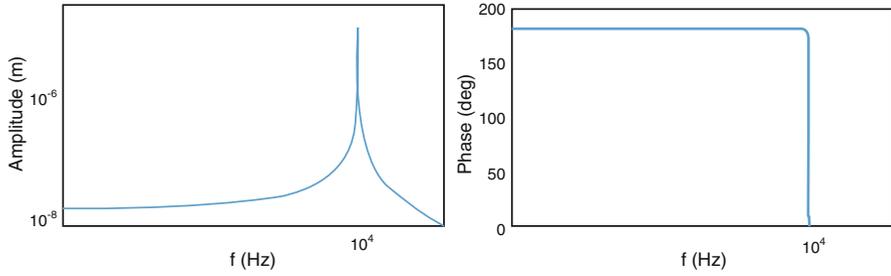


Fig. 7 Transfer function plots for amplitude and phase for frequency

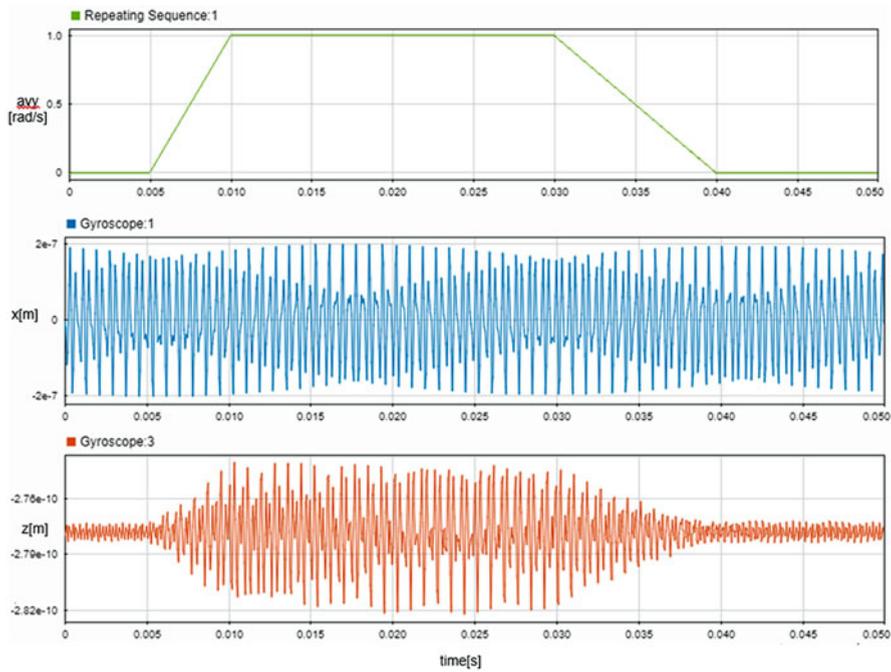


Fig. 8 Transient analysis for modified MEMS Gyroscope sensor

anchored at ends. Response characteristic improved meaningfully and is shown in Fig. 8.

3 Conclusions

Presented results are some of many outputs from MEMS vibrating gyroscope simulations which were used for evaluation and modification device. Programming

interface between MATLAB environment and Coventor MEMS+ gives us possibilities to extract much more results than from Coventor software itself and include automate parametrization and sweep analysis in loop(s), which is meaningfully time effective. Moreover, large library of different functional blocks included in Simulink, and possibility to import .3dsch file as fully functional block, allows to create complete system with external functional elements like sources or noises.

Presented environment is very convenient to improve performance of MEMS motion devices and allows to optimize their structure with included simulation options like eigenmode analysis, AC analysis, DC analysis, and sensitivity analysis. Some interesting results apply: AC rotation response, AC displacement response in frequency domain, acceleration sensitivity analysis and sidewall angle effect on Coriolis response, frequency variation with beam length, and electrostatic spring softening in relation to different geometrical dimensions of MEMS model.

Ground Vehicle Suspension Optimization Using Surrogate Modeling



Jeremy Mange

1 Introduction

Modeling and simulation (M&S) are critical tools for analysis of ground vehicles that allow for evaluating expected performance over a variety of tasks much more efficiently and cost-effectively than physical testing. In particular, high-fidelity physics-based M&S, such as the CREATE-GV Mercury tool [1] used within this study, enable accurate analysis of a number of ground vehicle mobility metrics. However, these types of tools tend to be computationally intensive, which limits their usefulness for processes that require hundreds or thousands of sequential simulations, as is the case for subsystem optimization.

One common approach to addressing this problem involves the use of *surrogate models*, simplified models that capture the important responses of the underlying system without the need for full system simulation. Creating surrogate models that are both accurate and computationally cheap is a difficult task that is highly context-specific. In this paper, we examine such a model for the optimization of ground vehicle suspension spring coefficients.

The organization of the paper is as follows: in the “Surrogate Models” section, we define the problem and the details of the surrogate models used. In the “Optimization Approach” section, we formulate the study as an optimization problem and discuss the optimization algorithm and parameters used. In the “Results” section, we present the results of the study and compare the results of the optimization using the surrogate models to a traditional optimization approach. Finally, in the “Conclusion and Future Work” section, we discuss the interpretation of the results and present related work to expand upon the concepts of this study.

J. Mange (✉)

US Army – CCDC GVSC – Analytical Methods, Warren, MI, USA
e-mail: jeremy.mange.civ@mail.mil; jeremy.b.mange.civ@mail.mil

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_65

887

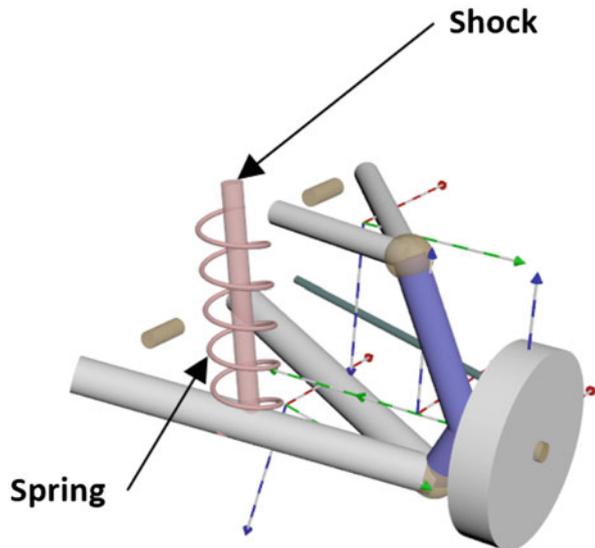
2 Surrogate Models

For this study, we modeled a notional ground vehicle of the general type of interest for our research context, namely, US Army-wheeled ground vehicles. For the sake of ease and clarity within the optimization, this model includes a simplified suspension in which the primary spring uses a single coefficient to model its behavior (see Fig. 1). The metric of interest for this study is the ISO 2361-5 static spine compressive stress (S_e) experienced by the driver during a large half-round shock event [2, 3], which is a measure of how “rough” of a ride the driver feels when going over a significant bump.

For this study, we employed the response surface methodology (RSM), which involves a design of experiments to construct a response surface, which is then used to approximate the results of the underlying model. For the construction of our surrogate models, we uniformly sampled a series of points over the search space for the optimization and ran a full high-fidelity system simulation of a half-round event, using the CREATE-GV Mercury software tool, to calculate the S_e value for each of those points. We created three surrogate models: a low-fidelity surrogate model consisting of only 10 points, a medium-fidelity surrogate model consisting of 100 points, and a high-fidelity surrogate model consisting of 1000 points. When using each of these surrogate models, results were calculated by linearly interpolating between points.

The relationship between the suspension spring coefficient and the S_e compressive stress value is highly non-linear, as shown in Fig. 2. This illustrates why optimization is both desirable and difficult for suspension components of this type—desirable because it is difficult for a human to predict what characteristics would

Fig. 1 Simplified ground vehicle suspension model [4]



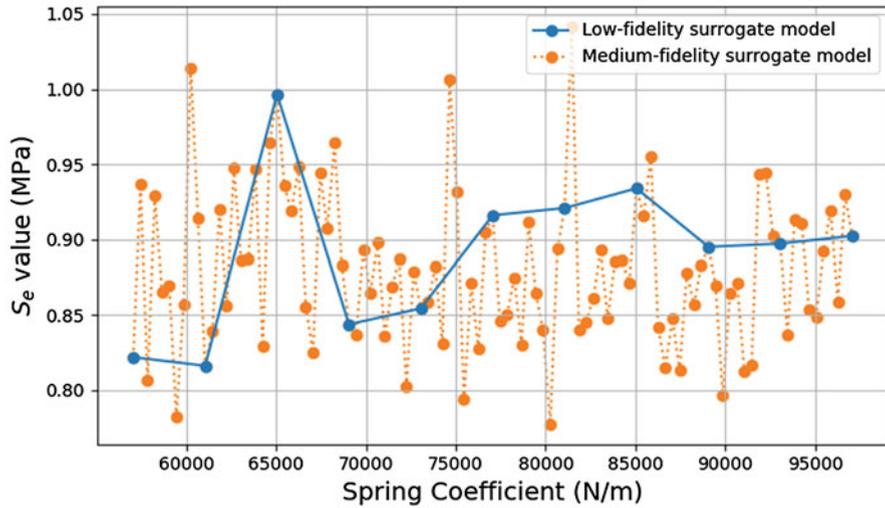


Fig. 2 Relationship between spring coefficient and S_e value

be ideal for driver safety and comfort as captured in the compressive stress metric and difficult because there are many local minima which can pose optimization challenges, particularly for derivative-based optimizers.

3 Optimization Approach

The formulation of this problem as an optimization task is fairly straightforward. We implemented a one-dimension, single-objective optimization problem wherein the spring coefficient for the ground vehicle suspension is the variable, and the S_e compressive stress metric value resulting from a simulated shock event, given a suspension with that spring characteristic, forms the objective function.

Because the relationship between the variable and objective function values is highly non-linear, we chose to use a derivative-free optimization algorithm. We chose a Particle Swarm Optimization [5] optimizer, using the default parameter settings from the PySwarm [6] Python implementation. Investigation of parameter values and alternative optimization algorithms is outside the scope of this study, as we are primarily interested in the accuracy of the optimization over the surrogate models compared to a traditional optimization, rather than the details of the algorithm itself. We constrained the spring coefficient to a reasonable range, given the suspension characteristics of the modeled notional vehicle, in order to define the search space for the optimization.

In order to test the constructed surrogate models, we ran a series of optimizations with a series of iteration limits, using each model to create a surrogate objective

function. Since Particle Swarm Optimization is a stochastic algorithm, we sampled each of these test points 100 times and averaged the results. We then compared the predicted S_e value for the suspension optimized using each surrogate model, to the actual S_e value for that spring coefficient. These results are presented in the following section.

4 Results

First, we compare the predicted optimized results from each of the surrogate models with the results of a traditional optimization, which is shown in Fig. 3. Recall that the low-fidelity surrogate model uses 10 sampled results, the medium-fidelity model uses 100, and the high-fidelity model uses 1000. Interestingly, the medium-fidelity surrogate model predicted slightly better (i.e., lower) S_e values than did the high-fidelity surrogate model, which may be due to the extremely noisy nature of the underlying objective function. However, these are simply the predicted values from the surrogate models; in order to truly test the accuracy of these models, we next turn to the true S_e values calculated from a full system simulation for each of the optimized spring coefficients from the surrogate models.

Figure 4 shows the true, full system simulation calculated S_e values for each of the optimized spring coefficients using the surrogate models. These were calculated by taking the average spring coefficient value at each number of objective function evaluation limits and running a complete CREATE-GV Mercury simulation with the notional vehicle using a suspension with that spring coefficient and using the

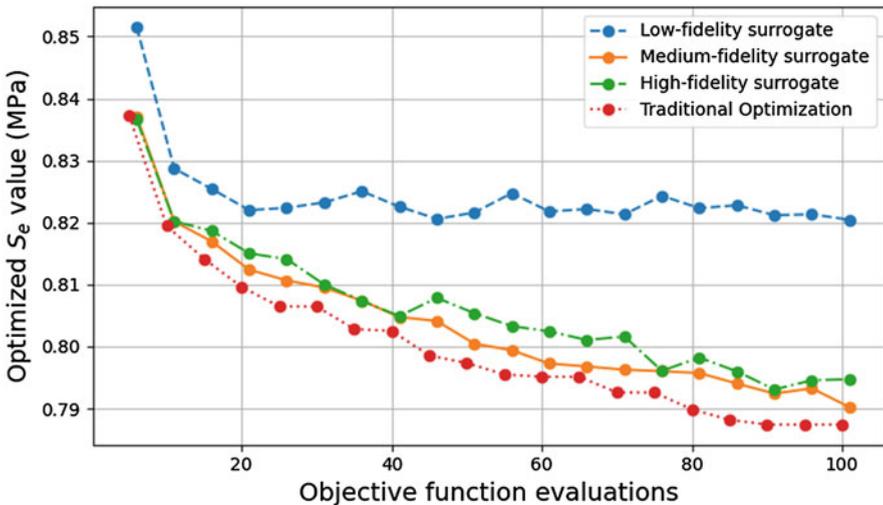


Fig. 3 Comparison of surrogate model-predicted values with traditional optimization

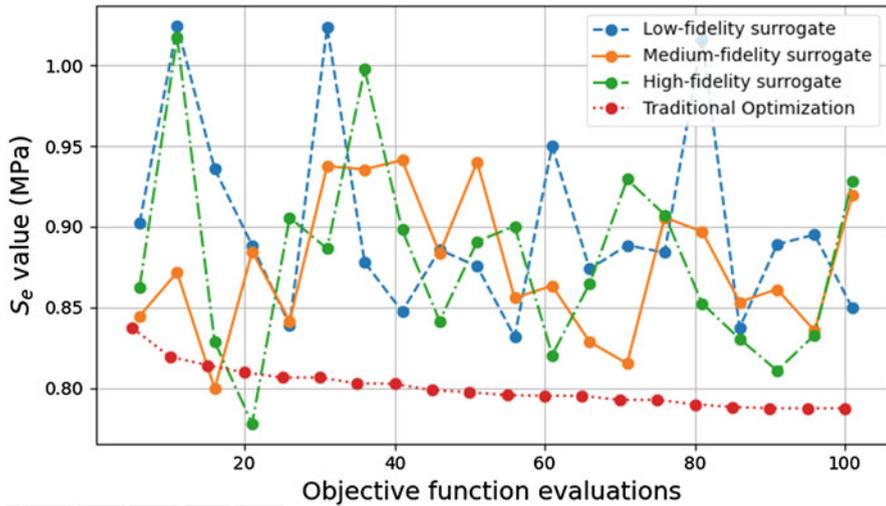


Fig. 4 True S_e values for surrogate model optimized spring coefficients

resulting S_e compressive stress value. As can be seen, these values are much noisier than the values predicted by the surrogate models themselves. While they do follow an overall downward trend as desired from the minimization optimization procedure, the values are so noisy as to be unsuitable for an actual optimization of this kind.

5 Conclusion and Future Work

The original intent of this study was to determine the suitability of fairly simple surrogate models for suspension optimization in a ground vehicle and further to compare the fidelity of several of these surrogate models with respect to the optimized values they produce. The optimization procedure was somewhat successful in demonstrating the fidelity level of interest—namely, that a medium-fidelity model consisting of 100 points was sufficient for the search space of interest for the suspension characteristics being optimized and in fact slightly outperformed a higher-fidelity model consisting of 1000 points. However, the underlying objective function used for this optimization was so noisy that the resulting predicted optimized values did not match well to the actual simulation system outputs, which undercut the viability of the optimization approach.

The general framework of this study is likely still useful for less noisy objective functions, of which there are many of interest in the ground vehicle domain. For future work, we would like to apply these same principles and approach to another such optimization tasks, to evaluate whether the surrogate model approach can be

proven effective. In addition, this would help validate the conclusions from this study regarding the appropriate fidelity of surrogate models for specific systems.

Beyond investigating different objective functions, it would be of interest to explore the sensitivity of this procedure to the choice of optimization algorithms, as well as parameters for a particular algorithm. The central concept of this study and the initial results are compelling, and additional work could bear out its usefulness for optimization of actual subsystem components within ground vehicles.

References

1. C. Goodin, J. Mange, S. Pace, T. Skorupa, D. Kedziorek, J. Priddy, L. Lynch, Simulating the mobility of wheeled ground vehicles with mercury. *SAE Int. J. Commer. Veh.* **10**(2017-01-0273), 498–502 (2017)
2. International Standards Organization, Mechanical vibration and shock—Evaluation of human exposure to whole-body vibration—Part 5: Method for evaluation of vibration containing multiple shocks. *ISO 2631-5:2018* (2018)
3. N. Alem, E. Hiltz, A. Breaux-Sims, B. Bumgardner, *Evaluation of New Methodology for Health Hazard Assessment of Repeated Shock in Military Tactical Ground Vehicles*. Army Aeromedical Research Lab Fort Rucker AL (2004)
4. Project Chrono, *Chrono::Vehicle Reference Manual*. Project Chrono—An Open Source Physics Engine (2020). http://api.projectchrono.org/manual_vehicle.html
5. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4 (IEEE, Piscataway, 1995), pp. 1942–1948
6. L. Miranda, PySwarms: a research toolkit for particle swarm optimization in Python. *J. Open Source Softw.* **3**(21), 433 (2018)

Part IX
Modeling, Visualization, Computational
Science, and Applications

Enhanced Freehand Interaction by Combining Vision and EMG-Based Systems in Mixed-Reality Environments



Carol Naranjo-Valero, Sriram Srinivasa, Achim Ebert, and Bernd Hamann

1 Introduction

As the adoption of mixed reality increases rapidly, the need to develop and improve more natural and intuitive interaction models gets more and more relevant. Consequently, new and more robust paradigms for improved freehand gestures as a natural form of interaction require further development and evaluation. A traditional approach to enable such interaction is the usage of vision-based systems. However, other alternatives, such as wearables based on EMG data, are often overlooked even though they offer promising opportunities for overcoming some of the limitations inherent to the vision-based systems and help to provide a more robust and flexible interaction that can be ultimately more intuitive.

With the development of more advanced mixed-reality platforms, such as the Microsoft HoloLens, the users do not only expect the ability to see virtual elements in coherence with the real space, but they also want to interact directly with such elements in a more natural way. They expect to approach a virtual model on top of a real table, touch it, and manipulate it as they would do with any other real object in their environment.

Most platforms rely only on their vision systems to provide freehand interaction. In order to guarantee an acceptable robustness and performance in real time, the result is often a very limited set of gestures that may not be the best fit for the different tasks that a user expects to accomplish in a mixed-reality environment. In

C. Naranjo-Valero (✉) · S. Srinivasa · A. Ebert
University of Kaiserslautern, Kaiserslautern, Germany
e-mail: valero@cs.uni-kl.de; ssriniva@cs.uni-kl.de; ebert@cs.uni-kl.de

B. Hamann
University of California, Davis, CA, USA
e-mail: hamann@cs.ucdavis.edu

the case of the HoloLens, the possible gestures for interaction are essentially two (bloom and air tap) plus the tracking of hands as included in the latest software update for the first generation of the HoloLens.

On the other hand, some wearable devices such as smartwatches that provide orientation data from inertial measurement units (IMU) have been used as interfaces in different applications and gained significant attention in HCI [1]. However, wearables based on electromyography (EMG) sensors or surface electromyography (sEMG) are still not explored enough. They have been long studied before but mostly for medical applications only, yet their potential in the area of human-computer interaction and mixed reality still requires further exploration and evaluation. The Myo armband by Thalmic Labs¹ has been so far the first commercial wearable and the most complete solution in this category [2] containing an array of eight surface EMG sensors and a 9-DOF inertial measurement unit, plus a proprietary firmware that offers the recognition five different gestures. Therefore, in this work we propose a new interaction model that takes advantage of the Myo armband technology in order to overcome limitations inherent to the vision system of the HoloLens.

Since the Myo armband does not require the gestures to be made in front of a camera, it poses as a suitable complement to the HoloLens: it offers an alternative for interaction in conditions where the HoloLens would not be able to respond, such as occlusion of the hands, or in the case where the hands are out of its field of view (FOV) [3, 4]. In this work we also show how expanding the set of gestures by integrating the Myo gestures in combination with the ones from the HoloLens provides higher flexibility and more intuitive interaction possibilities.

We have applied our interaction model for the case of 3D object manipulation given that many tasks in virtual and augmented reality involves the ability to manipulate 3D models in the environment [5]. For this purpose, we support the basic transformations for 3D rigid bodies: translation, rotation, and scaling.

We describe our use-case scenario used for the evaluation of our system, and we show that (1) our interaction model is more intuitive than the standard model based on the vision system of the HoloLens, (2) that providing a higher flexibility reflects in a better performance overall given a longer task, (3) and that although the performances of new introduced gestures based on EMG were less efficient than those based on the vision system, the preference of the users still favored the intuitiveness when allowed to choose. We conclude with a discussion of advantages, potentials, and limitations of this interaction paradigm as well as of the EMG as a growing technology in interaction systems.

¹The Myo armband was discontinued in 2018; however the existing units are still being supported. <https://bit.ly/2OErQRN>.

2 Related Work

Freehand Interaction in Mixed Reality Freehand interaction and mid-air gestures are often recommended when designing natural interfaces [6, 7] since they are intuitive, as they can relate better with interactions with the physical world, and are less cumbersome as they don't need controllers to be held. When used in mixed-reality systems, this interaction style is most commonly supported by vision systems, either with visible-light cameras or infrared cameras [6]. Among the methods for visual classification of hand gestures, the usage of deep learning models is reporting the most outstanding results [8–10], and among those, the methods incorporating fusion techniques combined with mesh-smoothing optimizations report the best results [11], indicating a strong advantage for hardware setups with multiple cameras of the same or different type. However, an ever present constraint of the state-of-the-art methods is the need of considerably high memory and processing capabilities, which is not the case for the main platforms for mixed-reality running on limited embedded cores.

The first generation of the Microsoft HoloLens, for instance, handles these constraints by providing a very limited set of gestures for interaction that are easily classified and therefore provide higher reliability and robustness for the user experience [12].

Previous work has tried to enhance the inherent interaction model of the HoloLens by combining with other user interfaces such as smartwatches or most commonly the leap motion. However, none of the recent work has studied the fusion with EMG wearable solutions.

EMG-Based Interaction We have found multiple examples of EMG-based interfaces [13–16] including Myo armband that show an increasing interest in such technology for HCI, although still in a premature phase. The Myo armband by Thalmic Labs is the most popular wearable of its kind for commercial use given its ergonomic and adaptive design and the pre-trained model that already offers the possibility to recognize five different gestures as shown in Fig. 1. Although the Myo has been used as interface for different systems [1, 17–19], its usage in the field of interaction for mixed reality is still not studied enough. For instance, we found very few examples where the hardware setup combines the HoloLens (being the most complete headset in this category) and the Myo on the same platform, but none of them is oriented to improve the interaction paradigm for mixed-reality environments. In the case of wekit [14], the Myo has been used as an input device to collect data as part of their expert elicitation system, rather than as a user interface. In another example [17] the Myo is used to control the velocity of a mobile robot; however, the interaction is rather limited and independent from the mixed-reality system itself, as it only acts as a binary command to increase or reduce the velocity of the robot. In a most recent work, we found that the Myo armband together with the HoloLens has been used as a training system for amputees' prostheses [1, 2]; however, the mapping of gestures during such training by its nature is not designed to be intuitive; in this case the mapped input do not correspond to the

visual feedback in the augmented reality environment. For example, the muscle activity corresponding to a fist in the forearm is mapped in the environment as a fingertip action [20].

Due to the nature of the sEMG technology, it is important to notice that the muscles that can be measured are mostly in charge of the flexion, extension, adduction, and abduction movements of the hands, plus the pronation and supination of the forearm, rather than the control of individual fingers, which are mainly controlled by muscles in the intermediate and deep layer [2]. Only the flexor digitorum superficialis is directly connected to the phalanges in the hand, but even this muscle is barely present on the superficial layer and is often considered only in the middle layer of the forearm muscles [21]. The intermediate and deep layers of muscles are where the main muscles that control the movements of the phalanges are present; unfortunately for the surface EMG sensors, the signals from such muscles are hardly detected in the superficial layer. This poses additional challenges for the recognition of gestures and movements based solely on EMG data, although recent research has shown interesting results using deep learning techniques [20, 22, 23]; however this is still a work in progress, and by the time being, we have to consider that the number and type of gestures recognized in a reliable manner are rather limited for practical applications.

In the future however, we should also consider that implantable solutions such as IMES [24], among others, are expected to be easier accessed and will probably be more accepted by users and society as the technology advances and gets more attention by researchers in HCI.

3 Design and Implementation

Our interaction model is designed to fulfill three main objectives. First, to provide a fluid interaction even when the model manipulated is outside of the field of view of the HoloLens; second, to reduce users' fatigue by identifying and integrating the most comfortable arm and hand positions for extended use; and third, to provide a bigger set of gestures in the context of 3D object manipulation that ultimately allow the user the choice for the most intuitive interaction.

3.1 Design Considerations

Our proposed interaction takes into account design guidelines from previous work in hand gesture design [7, 25, 26] as well as previous studies where the fatigue caused by some hand and forearm postures has been evaluated [25, 26]. In particular, with the usage of the armband as additional interaction, we are minimizing effects like the monkey syndrome identified in the previous studies.

Also, in order to provide a natural interaction, we also take into account some daily natural actions, such as “grabbing and object” (fist) or “turning a screwdriver” (forearm pronation) that can be mapped into our interaction model.

We have decided to include only a subset of Myo gestures to complement the HoloLens interaction. These are the fist, the spread-fingers, and the double-tap or pinch gesture as shown in Fig. 1. The wave-in and wave-out gestures were discarded since they have been identified in a previous study as some of the most cumbersome gestures that can cause fatigue for extended usage [26], while the fist is the gesture chosen to be held on for most of the interaction with the Myo armband for two main reasons: the first one being its identification as one of the most comfortable gestures in the same study and second because for the case of 3D object manipulation, it emulates the “grabbing” gesture that any user is already familiar with in a real environment, making it the most natural choice for such interaction. Finally, the spread-fingers and the double-tap are used only for brief periods of time. In addition, we use the orientation vector provided by the inertial unit in the armband in combination with each of the gestures while transforming the 3D objects in the scene.

3.2 Interaction Model

In our model, we reuse the interaction mapping from the Two Hand Manipulatable script of the Mixed-Reality Toolkit also called HoloToolkit [27] and integrate in parallel our Myo-supported interaction model as shown in Fig. 2, thus providing a permanent alternative of interaction for each transformation that responds better to the personal preferences of the users.

In the standard HoloToolkit interaction, the gestures use both hands for the case of rotation and scaling and one or two hands for translating. In all cases the air tap gesture is used.

In the proposed model supported by the Myo armband, we have favored the fist gesture for its usage in all the transformations as it has been identified in the literature as one of the less fatiguing positions [26]. The final interaction model is summarized in Table 1.



Fig. 1 Set of available gestures from the Myo armband

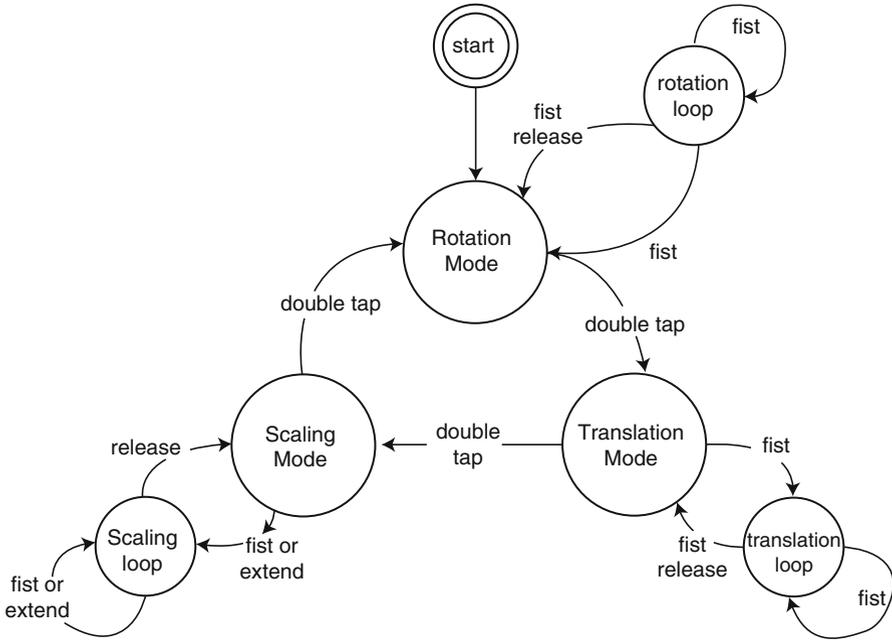


Fig. 2 Myo-based interaction model. The double-tap gesture is used to switch between transformation modes, while the fist is used as a common gesture to apply the corresponding transformation. In the scale mode, the “extend-fingers” gesture is included as an opposite gesture to the “fist” for scaling up and down, respectively

Table 1 Interaction modes included in our system

Operation	Interaction	
	HoloLens-only	Myo-supported
Translate	Hold air tap + release	Hold fist and release
Rotate	Two-hands air tap	Fist and rotate forearm
Scale	Two-hands air tap	Fist and spread-fingers

3.3 System Architecture

The application was built based on a client-server architecture as shown in Fig. 3. The server was used as bridge and parser for the data coming from the Myo, including the recognized standard gestures and the stream of orientation data from the embedded IMU. Here we used the MyoSharp library [28] as base for the Bluetooth protocol management. The data packages are formatted as json and sent via UDP.

The client running on the HoloLens receives the parsed data via UDP, where we reused the components provided by Baytas [29], and implements the additional gesture mapping as defined in our interaction model in Fig. 2. With this configuration, the original interaction model (TwoHandManipulatable) from the HoloToolkit [27]

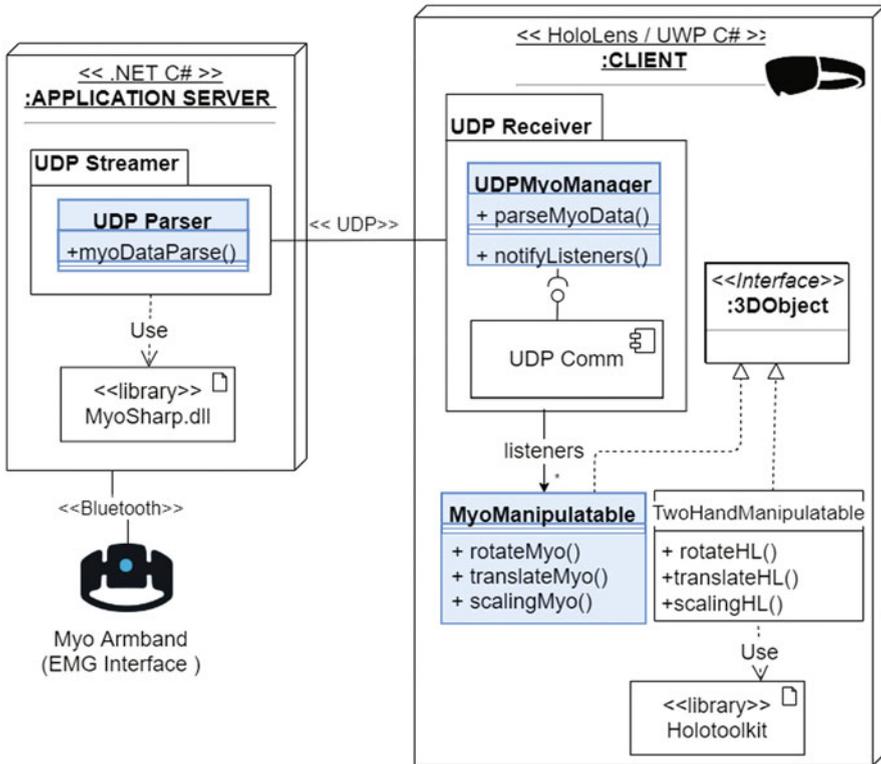


Fig. 3 Architecture diagram. The highlighted elements are the main components developed for the seamless integration of our new interaction techniques with the already existing ones from the HoloToolkit by Microsoft (Two-Hand Manipulation)

runs in parallel with the additional integrated gestures, therefore providing the users with a more flexible model that can respond to individual preferences.

4 Preliminar User Study

4.1 Use-Case Scenario

The implemented system supports the basic transformations for rigid objects in a 3D scenario, i.e., translation, rotation, and scaling. As one example for evaluation, we select a scenario where we display a simple model of a rocket, consisting of four pieces: engine mount, body tube, body cap, and nose cone as illustrated in Fig. 4.

The experiment was conducted in a room with dimensions of 3m × 4m where the 3D objects to be manipulated were distributed in a semi-random way. The

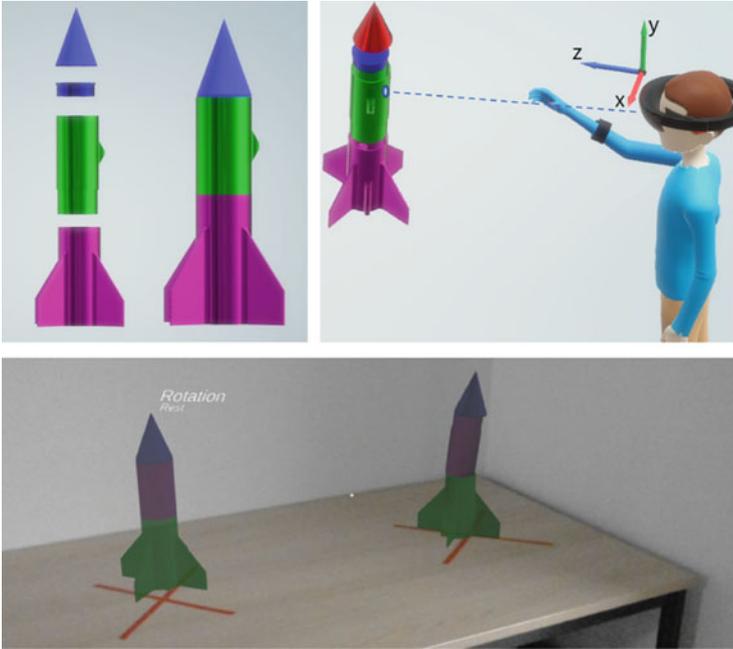


Fig. 4 Case scenario depiction. *Top-left*: 3D model of basic rocket taken from the standard 3D library of Microsoft. *Top-right*: Scene showing coordinate system from user perspective. The 3D transformations are applied to the target object determined by gaze direction (in z-direction). *Bottom*: Usage example: Mixed-reality capture of a completed task

room included two tables separated by 1.5 m, where some targets were marked with regular tape as indication for the experiment (as shown in Fig. 4).

4.2 Experiment

The goal of this experiment is to determine the advantages and limitations of our proposal compared to the standard interaction model of the HoloToolkit, i.e., HoloLens-only. To do so we evaluate some aspects of usability such as user performance and acceptance for both the individual gestures and for the completion of a more complex task defined in advance in the context of 3D object manipulations.

The experiment procedure included three stages: (1) introduction and learning of the interaction models, (2) individual gesture evaluation, and (3) free interaction evaluation. For the second and third stage, we measured the user performance as the completion time for each task.

And finally, at the end of the experiment, the participants were asked to fill out a questionnaire to establish the demographics of the participants and assess the general user acceptance of the interaction model. The questionnaire includes questions, such as gender, age, background, and previous experience with XR devices and EMG devices. We then included some questions that specifically evaluated the level of comfort perceived for each of the gestures included in our interaction model. Participants were asked to answer the questions on a seven-point Likert scale (Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree). We further included some open questions concerning the preferred gestures for each operation and the reasons for it, and also the participants were invited to provide suggestions regarding the interaction model and other possible desirable additional gestures.

Introduction and Learning of the Interaction Models During this familiarization stage, each participant was introduced to the technology and interaction model, including a pre-calibration procedure for the Myo armband. We asked participants to perform each of the possible gestures applied to a single object in the 3D scene (a regular cone representing the top of the rocket model). They were allowed to experiment freely with the system for about 5 min in order to familiarize themselves with the hardware, gestures, and expected response of the system.

Gesture Evaluation In the second stage, we measured completion times for each of the gestures providing the following instructions:

- Rotate the engine mount (rocket base) 360° along the z-axis as defined by a user's line of sight.
- Translate the body tube (main cylinder) to a designated area.
- Up-scale the body cap (small cylinder) in size and down-scale it back to its original size.

For each task we asked participants to repeat the procedure, one time using only the standard HoloLens-based interaction (air tap and double hands) and another time using only the gestures supported by the Myo armband (double-tap, fist, and extend-fingers). The order of such tasks was not specifically given so the users could decide any order to perform each of them.

Free Interaction Task For the final stage, we asked participants to assemble the complete rocket model using the floating parts spread around the environment. We placed a copy of the rocket already assembled as reference on top of a table and asked participants to build their own model next to it on a marked spot. For this task the participants used any gesture of their choice. We recorded their choices.

Figure 4 shows an example of one of the finished tasks for the third stage of the experiment, where users were asked to assemble the rocket in a free manner, i.e., allowing them to choose which gesture to use at any given time.

5 User Study Results

The user study was conducted with 15 participants (12 males and 3 females), between the ages of 23 and 32 years. Their fields of study and expertise varied from software engineering and intelligent systems to physical biology, civil engineering, and commercial vehicle technology. Five of the participants had previously used an AR or VR device (Oculus Rift). The others had no prior experience with extended reality systems. No participant had used the HoloLens before. Only one participant had used the Myo armband before.

When comparing the completion time for each gesture (Fig. 5), we observed that in general the gestures supported as part of the HoloToolkit were more efficient and that those supported by the Myo armband were performed slower in average but also showed a higher variance between users suggesting that wearable interface-based EMG technology is more susceptible to the particular characteristics of each user and that a more thorough calibration process is required in order to get more consistent results among users, since factors such as the fat tissue and muscle strength play a significant role in the final robustness observed.

However, for the third stage of evaluation, we observed an improvement in the completion times for the overall task when using the combination of both interaction modalities according to the users' preferences (Fig. 5).

In regard to the user acceptance of the system, we analyze the responses in perceived comfort for each operation as well as the chosen gestures when performing the task of free interaction.

In Fig. 6 we observed that the HoloLens standard interaction supported by the HoloToolkit was preferred for the rotation and translation actions, but the scaling operation done with the Myo was perceived in general more comfortable over scaling with the HoloLens.

As observed in Fig. 7, most participants preferred to perform rotations with the Myo than with the HoloLens due to the degree of intuitive use. Rotating with the

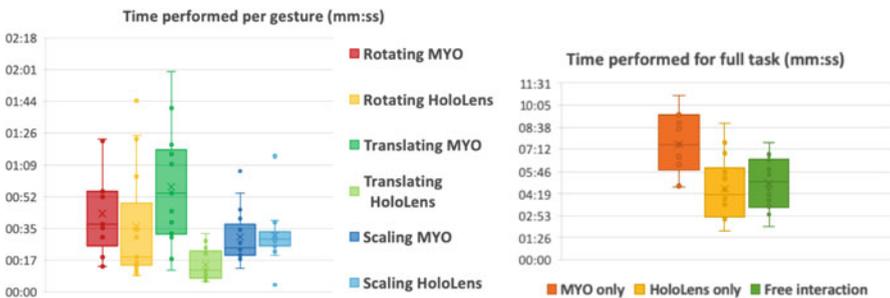


Fig. 5 User performance per gestures and full task. *Left:* Completion times for each gesture. *Right:* Completion times performing the assembly task with the different set of gestures

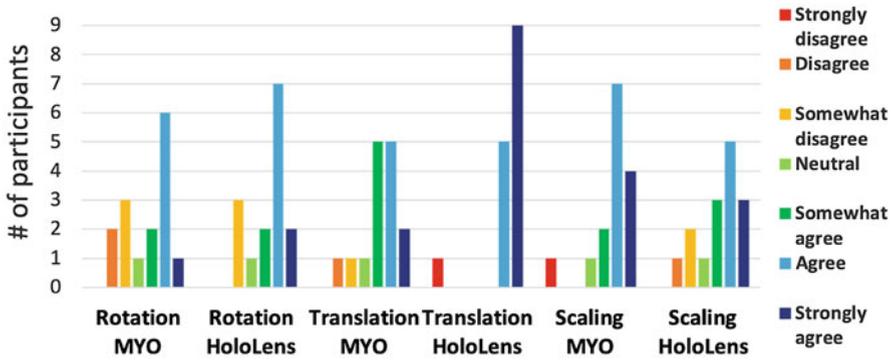


Fig. 6 Perceived comfort level per gesture: Agreement to the statement: “I found this gesture comfortable to use”

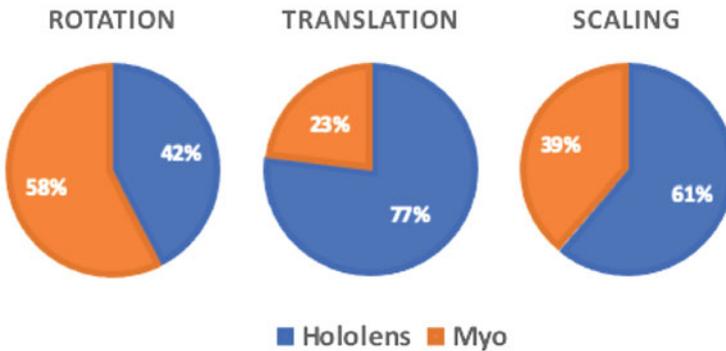


Fig. 7 Chosen gestures during free interaction

HoloLens required participants to use both hands, and sometimes the HoloLens did not recognize both hands.

Empirical Observations and Users’ Feedback Regarding the observations of the users during the evaluation, we collected several remarks. Some participants had problems with the weight of the HoloLens, making its use uncomfortable for them. The Myo armband was “too tight” for some participants, leaving indentations on their arms. Although the Myo did not make them uncomfortable, participants expressed relief when it was taken off. The application, for the most part, worked as intended. Despite some unexpected behavior seen during interaction, users were able to recover and continue with a task.

Scaling was executed more “smoothly” requiring the use of one hand only, a feature that users appreciated. The HoloLens required participants to use both hands in front of the camera to scale; participants found it difficult to execute tasks correctly.

Although the participants liked the idea of using the Myo to translate objects, the inaccuracy of translation made them prefer the use of the HoloLens to perform translation. Some participants used the Myo armband to translate objects over larger distances and the HoloLens to fine-tune final object placement. It is important to point out that during the evaluation some users experienced an unexpected artifact during interaction with the Myo, e.g., a temporary shaking movement of the 3D object being manipulated when releasing a fist gesture, as a result of noise present in the IMU data when switching to a gesture. We noticed that these artifacts were more severe for some users. We believe that this effect is due to noise in the signals generated by a device. However, we consider that this artifact did not affect our results, as only time was measured and the criteria for a completed task were established when a user was satisfied with the final position, rotation phase, and size of each of the 3D objects.

In the case of the Myo gestures, many participants expressed difficulty to perform the double-tap gesture, while for some others, it worked as intended every time. Although it was not our focus to test the robustness for the recognition of each gesture, we observed that the disparity of performance between users when using the Myo armband was reflected in a higher deviation standard in the completion times for individual gestures. Participants provided suggestions regarding the desirable additional gestures to be supported. For example, some users would like to use the wave-in and wave-out gestures of the Myo to move objects sideways, in addition to using just the existing translation capability. Some users indicated that they would prefer a gesture different from double-tap to switch modes for Myo manipulations as that gesture was not easily recognized.

5.1 Discussion

In our experiment, we investigated the user acceptance toward the usage of the proposed interaction model in comparison with the standard interaction capabilities of a vision-based system. We evaluated the attitude of the users toward our multimodal setup and tested the difference of performance when using different sets of gestures in order to complete a task fairly common in mixed-reality environments, e.g., applying rigid transformations to 3D objects in the scene.

We can conclude that users generally felt positive about the interaction paradigm implemented in our system and its potential. Our prototype interaction system was sufficient to demonstrate intuitive and adequate interaction for 3D object manipulations. Among some interesting remarks, we found that although some gestures supported by the Myo armband proved to be less efficient according to the measured performance (Fig. 5), the users still chose to use them when facing the free interaction task. Also important is that even though the HoloLens standard gestures performed overall better than the Myo-supported gestures, when using both systems in combination, the completion of the assembly task was more efficient,

proving then that the flexibility of interaction represented a significant advantage in the multimodal interaction model.

6 Conclusions and Future Work

We presented the design and implementation of an interaction model combining the advantages of a vision-based system, the most commonly one used for freehand interaction, with a wearable interface based on EMG sensors and inertial measurements. Our model can overcome several of the limitations typical for vision-based systems by allowing a fluid interaction even when hands are out of the camera's field of view (FOV) or occluded by other objects. Our approach is aiming at reducing fatigue of the upper arm as gestures do not need to be sustained in a high position to lie inside the FOV.

Our model offers a high degree of flexibility and intuitiveness for interaction in mixed-reality environments. We show that even when the time required to complete some tasks is longer when using Myo-supported gestures, users often preferred to use them as they are highly intuitive for 3D object manipulation. As part of our design considerations, we have discarded the use of sustained gestures identified as uncomfortable in the literature, e.g., wrist adduction or the extension called "wave-out" by Myo. However, our user study suggests that one should include these as well in a system, indicating users' preference to have available a larger set of gestures for more flexibility during interaction.

Concerning limitations for EMG-based interfaces, we conclude that a higher resolution is extremely important to support accuracy and robustness among a large set of users. Challenges of this technology include the need of a better calibration process for each user since the observed variance among users was significant. Ergonomic issues must also be addressed and becomes especially relevant when carrying on tasks for longer times. In addition we suggest the further study of implantable EMG solutions as its technology advances with adequate safety standards and becomes more accessible for the general public.

Acknowledgments Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—252408385—IRTG 2057

References

1. E.C.P.P Silva, E.W.G.G Clua, A.A. Montenegro, Sensor data fusion for full arm tracking using Myo armband and leap motion, in *2015 14th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)* (2016), pp. 128–134
2. P. Visconti, F. Gaetani, G.A. Zappatore, P. Primiceri, Technical features and functionalities of Myo armband: an overview on related literature and advanced applications of myoelectric armbands mainly focused on arm prostheses. *Int. J. Smart Sensing Intell. Syst.* **11**(1), 1–25 (2018)

3. O. Kreylos, HoloLens and Field of View in Augmented Reality (2015)
4. D. Ren, T. Goldschwendt, Y. Chang, T. Hollerer, Evaluating wide-field-of-view augmented reality with mixed reality simulation, in *Proceedings - IEEE Virtual Reality* (2016)
5. M. Krichenbauer, G. Yamamoto, T. Taketom, C. Sandor, H. Kato, Augmented reality versus virtual reality for 3D object manipulation. *IEEE Trans. Vis. Comput. Graph.* **24**(2), 1038–1048 (2018)
6. A. Theil Cabreira, F. Hwang, An analysis of mid-air gestures used across three platforms, in *Proc. 2015 Br. HCI Conf. - Br. HCI '15* (2015), pp. 257–258
7. R. Aigner, D. Wigdor, H. Benko, M. Haller, D. Lindlbauer, A. Ion, S. Zhao, J.T. K. Valino Koh, Understanding Mid-Air Hand Gestures : A Study of Human Preferences in Usage of Gesture Types for HCI. Microsoft Research Technical Report MSR-TR-2012-111 (2012)
8. P. Molchanov, S. Gupta, K. Kim, J. Kautz, Hand gesture recognition with 3D convolutional neural networks, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (2015)
9. J.S. Supančič, G. Rogez, Y. Yang, J. Shotton, D. Ramanan, Depth-based hand pose estimation: methods, data, and challenges. *Int. J. Comput. Vis.* **126**(11), 1180–1198 (2018)
10. S. Kolariä, A. Raposo, M. Gattass, Direct 3D manipulation using vision-based recognition of uninstrumented hands, in *Tenth Symposium on Virtual and Augmented Reality* (2008), pp. 212–220
11. J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff, A. Topalian, E. Wood, S. Khamis, P. Kohli, S. Izadi, R. Banks, A. Fitzgibbon, J. Shotton, Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Trans. Graph.* **35**(4), 1–12 (2016)
12. N. Chaconas, T. Höllerer, An evaluation of bimanual gestures on the Microsoft HoloLens, in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (IEEE, Piscataway, 2018), pp. 1–8
13. Y. Yang, S. Chae, J. Shim, T.-D. Han, EMG Sensor-based two-hand smart watch interaction, in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology—UIST '15 Adjunct* (2015), pp. 73–74
14. W. Guest, F. Wild, A. Vovk, M. Fominykh, J. Karjalainen, C. Smith, J. Rasool, S. Aswat, Affordances for capturing and re-enacting expert performance with wearables. *Eur. Conf. Tech. Enhanc. Learn.* **10474**, 403–409 (2017)
15. P. Paudyal, A. Banerjee, S.K.S. Gupta, SCEPTRE: A pervasive, non-invasive, and programmable gesture recognition technology, in *International Conference on Intelligent User Interfaces, Proceedings IUI* (2016), pp. 282–293
16. T. Peters, *An Assessment of Single-Channel EMG Sensing for Gestural Input* (Dartmouth College, Hanover, 2014), pp. 1–14
17. M. Wu, Y. Xu, C. Yang, Y. Feng, Omnidirectional mobile robot control based on mixed reality and sEMG signals, in *Proceedings 2018 Chinese Automation Congress, CAC 2018* (2019), pp. 1867–1872
18. E.J. Rechy-Ramirez, A. Marin-Hernandez, H.V. Rios-Figueroa, E.J. Rechy-Ramirez, A. Marin-Hernandez, H. Vladimir Rios-Figueroa, A human—computer interface for wrist rehabilitation: a pilot study using commercial sensors to detect wrist movements. *Vis. Comput.* **35**(1), 41–55 (2019)
19. U. Côté-Allard et. al., Towards the use of consumer-grade electromyographic armbands for interactive, artistic robotics performances, in *Proc. 26th IEEE Int. Symp. Robot Hum. Interact. Commun.* (2017), pp. 1030–1036
20. E. Ceolini, G. Taverni, L. Khacef, M. Payvand, E. Donati, Live Demonstration: sensor fusion using EMG and vision for hand gesture classification in mobile applications, in *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)* (2019), p. 5090
21. J. Oliver, Muscles in the posterior compartment of the forearm (2017)
22. A. Phinyomark, E. Scheme, EMG pattern recognition in the era of big data and deep learning. *Big Data Cogn. Comput.* **2**(3), 21 (2018)

23. X. Xi, C. Yang, J. Shi, Z. Luo, Y.-B. Bo Zhao, Surface electromyography—based daily activity recognition using wavelet coherence coefficient and support vector. *Neural Process. Lett.* **50**, 2265–2280 (2019)
24. M. Voelker, A. Nikas, H. Zhou, J. Hauer, R. Ruff, K.-P. Hoffmann, Implantable EMG measuring system, in *AMA Conferences 2015—SENSOR 2015 and IRS2 2015*, vol. 1 (2015), pp. 426–429
25. J.D. Hincapié-Ramos, X. Guo, P. Moghadasian, P. Irani, Consumed endurance: A metric to quantify arm fatigue of mid-air interactions, in *Proceedings of the Conference on Human Factors in Computing Systems* (2014), pp. 1063–1072
26. D. Rempel, M.J. Camilleri, D.L. Lee, The design of hand gestures for human-computer interaction: Lessons from sign language interpreters. *Int. J. Hum. Comput. Stud.* **72**(10–11), 728–735 (2014)
27. Microsoft, MixedRealityToolkit-Unity (HoloToolkit) v. 2017.4.3 (2017)
28. Thalmic Labs Inc., Myo Sharp—Github (2019)
29. M.A. Baytaş, HoloLens UDP repository (2017)

Parameterizations of Closed-Loop Control Systems would be perfectly fine



Cs. Bányász, L. Keviczky, and R. Bars

1 Introduction

A lot of parameterizations of control structures were investigated in our papers and books in the last years. Mostly the influence of a model used is also treated. In this chapter, the topologies based on Keviczky-Bányász (KB) and Youla parameterization are discussed [6–9].

The most important element of a closed-loop control system is the regulator C , which has to be determined during the design procedure [1]. Direct and indirect parameterization methods are reviewed below, which can help make this task easier.

The closed system is internally stable, and if given a bounded excitation at the arbitrary point of the system, the generated signals in any point remain bounded [2]. Thus, stable transfer functions must be obtained between any two input-output points. The mathematical condition of this property can be formalized in the simplest way by introducing the transfer matrix \mathbf{T}_t of the closed system, which represents the relationships between any two independent outer and two inner signals. A suitable choice for \mathbf{T}_t is

$$\mathbf{T}_t(P, C) = \frac{1}{1 + CP} \begin{bmatrix} P \\ 1 \end{bmatrix} [C \ 1] = \mathbf{D}^{-1} \mathbf{s} \mathbf{r}^T = \mathbf{s} \mathbf{r}^T \mathbf{D}^{-1} = \frac{1}{1 + CP} \begin{bmatrix} CP & P \\ C & 1 \end{bmatrix} \quad (1)$$

Cs. Bányász (✉) · L. Keviczky
Institute for Computer Science and Control, SZTAKI, Budapest, Hungary
e-mail: banyasz@sztaki.hu; keviczky@sztaki.hu

R. Bars
Budapest University of Technology and Economics, Budapest, Hungary
e-mail: bars@aut.bme.hu

where $\mathbf{s} = [P \ 1]^T$, $\mathbf{r} = [C \ 1]^T$ and $\mathbf{D} = (1 + CP)\mathbf{I}$. It can be seen that the control loop is internally stable if and only if $\mathbf{T}_t(P, C) \in s$, where s is the set of the CT stable linear processes. Using simple algebraic rearrangements

$$\mathbf{T}_t(P, C) = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{H}_t(P, C) + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tag{2}$$

is obtained, where

$$\mathbf{H}_t(P, C) = \begin{bmatrix} 1 & P \\ -C & 1 \end{bmatrix}^{-1} = \mathbf{D}^{-1} \begin{bmatrix} 1 & -P \\ C & 1 \end{bmatrix} \tag{3}$$

The stability of the transfer matrix $\mathbf{T}_t(P, C)$ can be investigated via the stability of matrix $\mathbf{H}_t(P, C)$. A similar statement can be made for the dual transfer function $\mathbf{T}_t(C, P) = \mathbf{r}\mathbf{D}^{-1}\mathbf{s}^T = \mathbf{D}^{-1}\mathbf{r}\mathbf{s}^T = \mathbf{r}\mathbf{s}^T\mathbf{D}^{-1}$ by introducing the matrix

$$\mathbf{H}_t(C, P) = \begin{bmatrix} 1 & C \\ -P & 1 \end{bmatrix}^{-1} = \mathbf{D}^{-1} \begin{bmatrix} 1 & -C \\ P & 1 \end{bmatrix} \tag{4}$$

2 Youla Parameterization

Let us find parameterization for the closed-loop which makes all elements of $\mathbf{T}_t(P, C)$ stable. Introduce the following transfer function as parameter

$$Q = \frac{C}{1 + CP} \tag{5}$$

by means of which the new form of the transfer matrix is

$$\mathbf{T}_t(P, C) = \mathbf{T}_t(P, Q) = \begin{bmatrix} P \\ 1 \end{bmatrix} [Q \ 1 - QP] = \mathbf{s}\mathbf{q}^T = \begin{bmatrix} QP & P(1 - QP) \\ Q & 1 - QP \end{bmatrix} \tag{6}$$

It is clear that for the stable process $P \in s$ and stable parameter $Q \in s$, all elements of \mathbf{T}_t are stable; thus, the internal stability of the closed loop is ensured (here $Q(\omega = \infty)$ is finite and regular). Otherwise, Q represents the transfer function of a one-degree-of-freedom control loop between the reference signal r and the actuating signal u . All elements of \mathbf{T}_t are linear (therefore convex) in Q . Similarly, the sensitivity functions are also linear

$$T = \frac{CP}{1 + CP} = QP; \quad S = \frac{1}{1 + CP} = 1 - QP \tag{7}$$

The above procedure is called Youla parameterization (YP), where Q is the Youla parameter. From (5), the Youla -parameterized regulator is

$$C = \frac{Q}{1 - QP} \tag{8}$$

The Youla-parameterized control loop is shown in Fig. 1, where the above notations are used.

The Youla parameter, as a matter of fact, is a stable (by definition), regular transfer function

$$Q(s) = \frac{C(s)}{1 + C(s)P(s)} \text{ or shortly } Q = \frac{C}{1 + CP} \tag{9}$$

where $C(s)$ is the stabilizing regulator and $P(s)$ is the transfer function of the stable process.

It follows from the definition of the Youla parameter that the structure of the realizable and stabilizing regulator in the Youla-parameterized control loop is fixed:

$$C(s) = \frac{Q(s)}{1 - Q(s)P(s)} \text{ or shortly } C = \frac{Q}{1 - QP} \tag{10}$$

The Youla-parameterized control loop is shown in Fig. 1. The sensitivity and complementary sensitivity functions linear in Q of the closed-control systems were defined by (7). It is interesting to observe that the YP regulator of (8) can be realized by a simple control loop with positive feedback as shown in Fig. 2.

The relationships between the most important signals of the closed system can be obtained with simple calculations

$$\begin{aligned} u &= Qr - Qy_n \\ e &= (1 - QP)r - (1 - QP)y_n = Sr - Sy_n \quad y = QPr + (1 - QP)y_n = Tr + Sy_n \\ y &= QPr + (1 - QP)y_n = Tr + Sy_n \end{aligned} \tag{11}$$

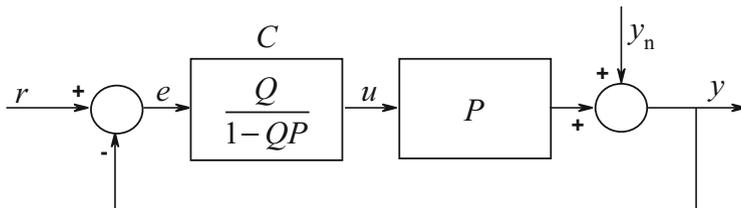
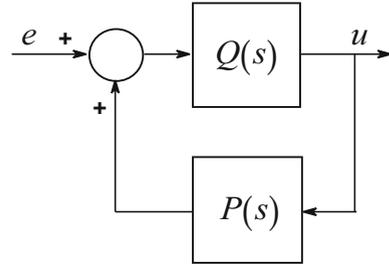


Fig. 1 Youla-parameterized control loop

Fig. 2 Realization of YP regulator



The effect of r and y_n on u and e is completely symmetrical (not considering the sign). The input of the process depends only on the external signals and $Q(s)$.

Now apply the inverse of Q , connected serial as in Fig. 3a, in the block diagram (Fig. 1) of the Youla-parameterized control loop. In this way, the tracking property becomes independent of Q , i.e., Pr' ; thus, the closed-loop seems to be formally opened. It can be easily checked that this block diagram is equivalent to Fig. 3b. Figure 3c shows the effects of the inputs on the output. The relationships between the most important signals are:

$$\begin{aligned} u &= r' - Qy_n \\ e &= 0 - (1 - Q)y_n = Sy_n \\ y &= Pr' + (1 - QP)y_n = T'r' + Sy_n \end{aligned} \tag{12}$$

The Youla parameterization extended by Q^{-1} , which formally opens the closed loop, is called KB parameterization (after the authors Keviczky and Bányász [3–5]). As regards the operation of the KB-parameterized loop, it should be noted that here the reference signal has a direct effect on the input of the process, and thus it does not go through the regulator and the whole closed loop. All further control effects (regarding the reference signal) are in operation only when the inner model is not equal to the real process. It can be seen from Fig. 3b that the KB parameterization is independent of the Y parameterization concerning the reference signal effect, since it opens the closed-loop for any arbitrary regulator C , and thus the overall tracking transfer function is always P . The effect of the disturbance signal, however, can be compensated only by the Y parameterization via a simple linear transfer function $(1 - QP)$ linear in Q . Thus, in the case of two-degree-of-freedom systems (TDOF), the two principles have to be jointly applied.

Using the scheme in Fig. 2, the Youla-parameterized closed-loop control shown in Fig. 1 can be redrawn to the equivalent form of Fig. 4. This classical internal model-based scheme is called the *Internal Model Control: IMC* [9]. The basic principle of this control is that it has feedback only from the deviation (ε) between the process output and the model output to create the error signal of the control. This error signal is zero in the ideal case when the internal model is completely equal to the process. This case is shown above. But in reality, the internal model $\hat{P}(s)$ is only a good approximation of the true process $P(s)$, since the original system is not

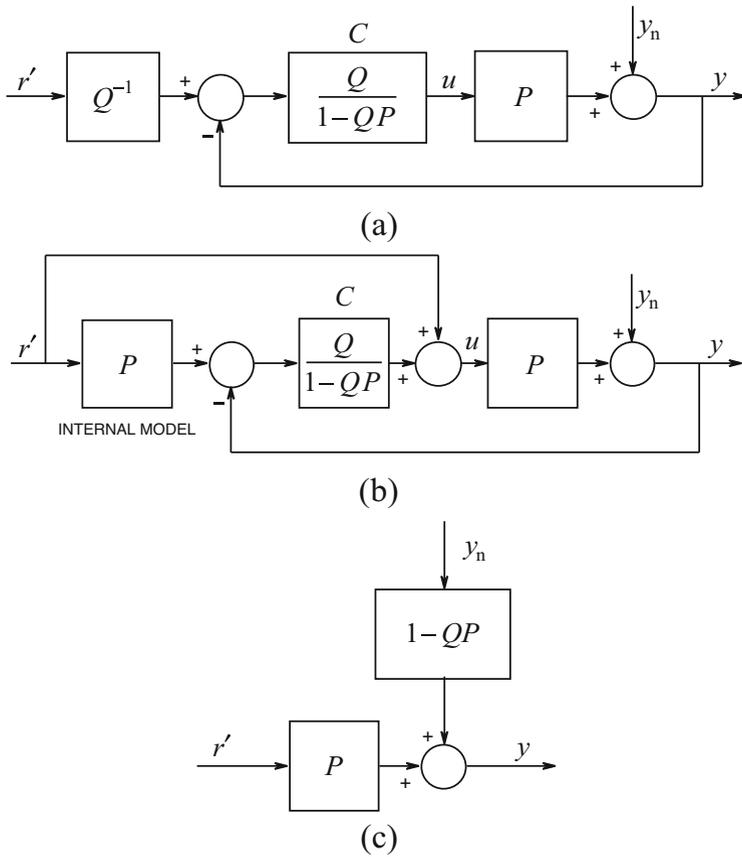


Fig. 3 Block diagrams for opening the closed-loop control systems

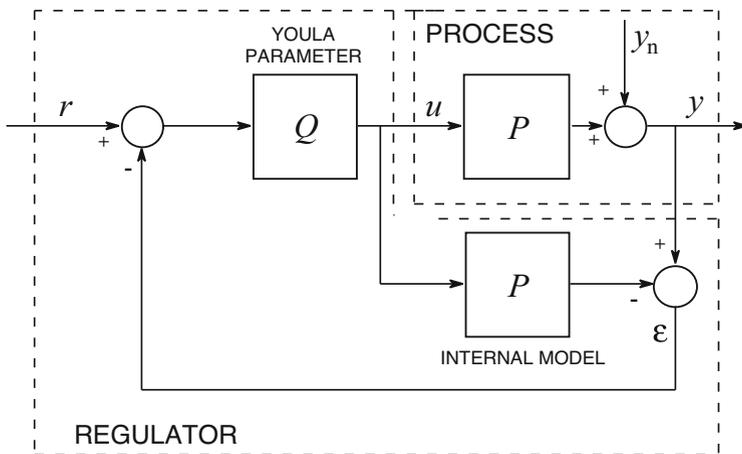


Fig. 4 Equivalent IMC loop

known exactly. For the sake of simplicity, only the ideal case is discussed here first. The relationships between the most important signals of the closed system can be obtained with simple calculations

$$\begin{aligned} u &= Qr - Qy_n \\ y &= QPr + (1 - QP) y_n = Tr + Sy_n \end{aligned} \tag{13}$$

i.e., the same as in (11).

3 Keviczky-Bányász Parameterization

As mentioned previously, the Youla parameterization extended by Q^{-1} , which formally opens the closed-loop, is called KB parameterization after the authors (Keviczky and Bányász [3–5]) who suggested the method. The basic scheme of the KB parameterization is shown in Fig. 5.

The basic relationships of the closed-loop are

$$y = (1 - QP + QQ_{KB})|_{Q_{KB}=P} Pr' + (1 - QP) y_n = Pr' + (1 - QP) y_n \tag{14}$$

It is worth noting that the reference signal takes effect directly on the input of the process, and thus it does not go through the regulator and the closed loop. The controlling effect regarding the reference signal operates only if the internal model is not equal to the real process. Considering the reference signal effect, the KB parameterization shown in Fig. 5 is independent of the Y parameterization since it opens the closed loop even for an arbitrary regulator C , i.e., the overall tracking transfer function is always P if $Q_{KB} = P$ is chosen. The effect of the disturbance signal is compensated only via the simple transfer function $(1 - QP)$ linear in Q . Thus, in the case of $TDOF$ systems, the two principles have to be applied simultaneously. If $Q_{KB} = P$ is chosen, the equivalent closed loop corresponds to the open loop in Fig. 6.

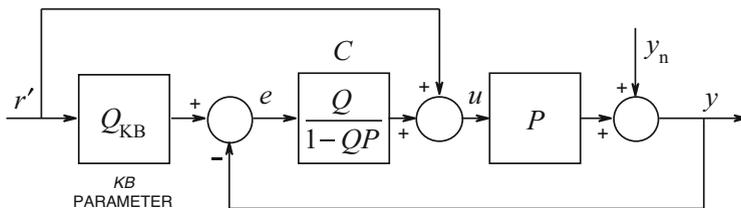


Fig. 5 Scheme of the KB parameterization

Fig. 6 The ideal choice of the KB parameter $Q_{KB} = P$ opens the system

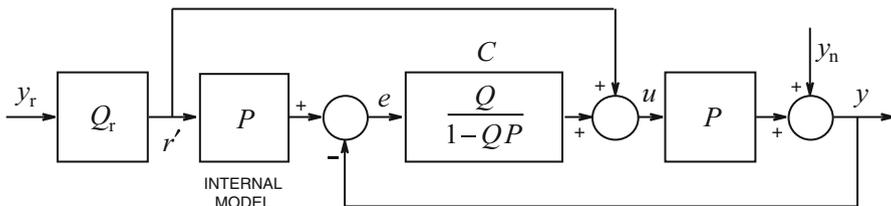
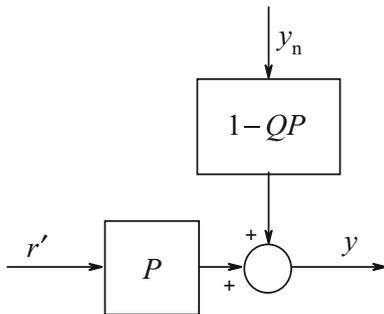


Fig. 7 Two-degree-of-freedom version of the YP control loop

Although the invention of the KB parameterization is inspired by the Youla parameterization, its validity is more general, since it opens the closed control loop for any regulator

$$\begin{aligned}
 y &= (1 + Q_{KB}C) \frac{P}{1+CP} y_r + \frac{1}{1+CP} y_n \\
 &= \frac{(1+Q_{KB}C)P}{1+CP} \Big|_{Q_{KB}=P} y_r + \frac{1}{1+CP} y_n = P y_r + \frac{1}{1+CP} y_n
 \end{aligned}
 \tag{15}$$

Furthermore,

$$u = y_r - \frac{C}{1+CP} y_n; \quad e = 0 - \frac{1}{1+CP} y_n
 \tag{16}$$

From the last equation in (11), it can be seen that both the Youla parameterization and the IMC have the transfer function QP_r concerning the reference signal tracking. If the KB parameterization introduced in the figures of Fig. 3 is applied, then the Youla parameterization can be simply extended for TDOF control systems. To do this, let us simply apply a parameter Q_r for the design of the tracking properties and connect it in serial to the KB-parameterized loop of Fig. 3, so the block diagram of Fig. 7 is obtained.

The overall transfer characteristics for this system are

$$\begin{aligned}
 u &= Q_r y_r - Q y_n \\
 e &= 0 - (1 - QP) y_n = 0 - S y_n \\
 y &= Q_r P y_r + (1 - QP) y_n = T_r y_r + (1 - T) y_n = T_r y_r + S y_n
 \end{aligned}
 \tag{17}$$

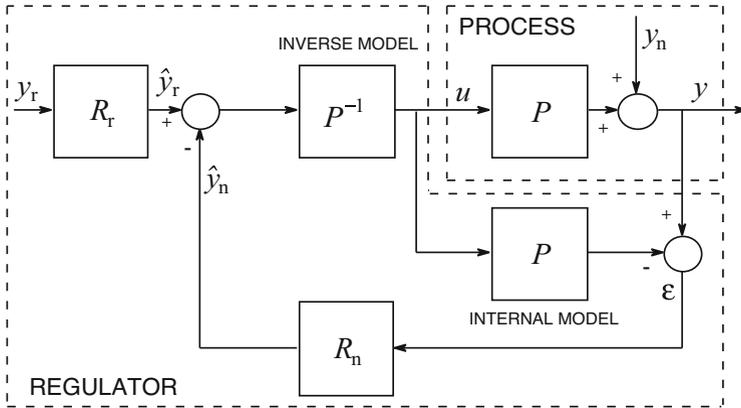


Fig. 8 Extension of the IMC-based ideal control loop with reference models

where the tracking properties can be designed by choosing Q_r in $T_r = Q_r P$ and the noise rejection properties by choosing Q in $T = QP$. These two properties can be handled separately. The reference signal of the whole system is denoted by y_r . The conditions for Q_r are the same as for Q . The meaning of T_r is analogous to the meaning of the complementary sensitivity function T of the one-degree-of-freedom control loop for tracking.

The IMC of Fig. 4 can be further developed according to Fig. 8. Here, the predicted value \hat{y}_n of the output disturbance y_n is constructed from the difference ε between the outputs of the process and the model by the predictor R_n . Similarly, the predictor R_r provides the predicted value \hat{y}_r of the reference signal y_r . The noise rejection is performed by giving the predicted value $-\hat{y}_n$ of the disturbance to the process input via the inverse of the process model; thus, in the case of accurate estimation, the disturbance is eliminated. The reference signal tracking operates in a similar way. Here, the operation of R_r can be considered a reference model (desired system dynamics) and, therefore, the introduced predictors are also called reference models. It is generally required that these predictors (filters) are strictly proper with unit static gain, i.e., $R_n(\omega = 0) = 1$ and $R_r(\omega = 0) = 1$.

By introducing the reference models (or predictors) R_r and R_n , the design of the regulator is simplified to the design of these goals instead of selecting Q_r and Q .

4 Model-Based Youla Regulator

Investigate first the model-based version of the topology given in Fig. 3b, i.e., as Fig. 9 shows.

Here, for the sake of simplicity, the influence of the output disturbance y_n is not investigated.

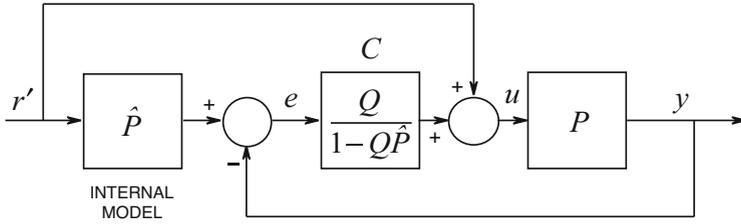


Fig. 9 The model-based Youla regulator with KB parameterization

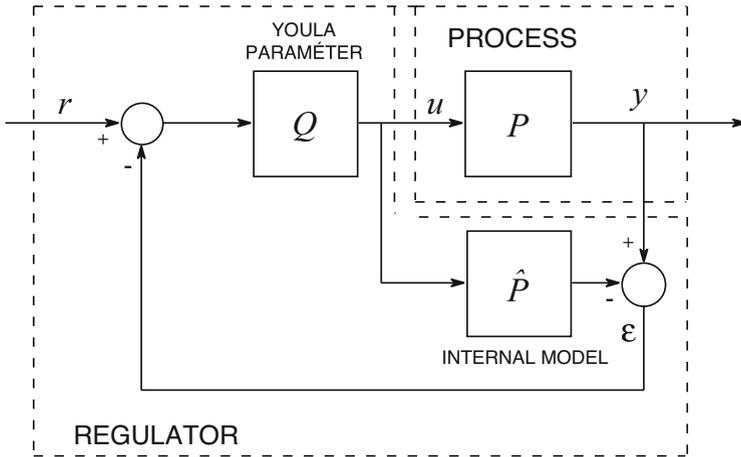


Fig. 10 The model-based IMC regulator

$$u = r' - Q\varepsilon \tag{18}$$

where

$$\varepsilon = y - \hat{P}u \tag{19}$$

is the model output error (see Figs. 4 and 8). The other important signals in the closed-loop are

$$\begin{aligned} e &= (1 - Q\hat{P})\varepsilon = \hat{S}\varepsilon \\ y &= \hat{P}u + \varepsilon = \hat{P}r' + (1 - Q\hat{P})\varepsilon = \hat{T}'r' + \hat{S}\varepsilon \end{aligned} \tag{20}$$

Next, investigate the model-based version of the IMC topology given in Fig. 4 as Fig. 10 shows.

Here, the process input and outputs are

$$\begin{aligned}
 u &= Qr - Q\varepsilon \\
 y &= \hat{P}u + \varepsilon = Q\hat{P}r + (1 - Q\hat{P})\varepsilon = \hat{T}r + \hat{S}\varepsilon
 \end{aligned}
 \tag{21}$$

From the above analysis, it can be well seen that the KB parameterization provides zero control error for the ideal case, when $\hat{P} = P$, i.e., the process is exactly known. In case of model error, the control error is $e = \hat{S}\varepsilon$, i.e., it will depend on the output error ε . The influence of this error can be properly attenuated if \hat{S} is selected well. \hat{S} is generally a high-pass filter, so the low frequency domain is always damped.

5 Application of the Observer Principle

A possible combination of the state-feedback principle [2] and the IMC regulator is shown in Fig. 11. Here, the influence of the output disturbance y_n is not investigated. The process input and outputs are

$$u = \frac{Q}{1 + QK_K P}r; \quad y = \frac{QP}{1 + QK_K P}r
 \tag{22}$$

The model output error ε is zero now, because $\hat{P} = P$, i.e., the process is exactly known. The resulting block scheme (Fig. 12) slightly differs from the original IMC system.

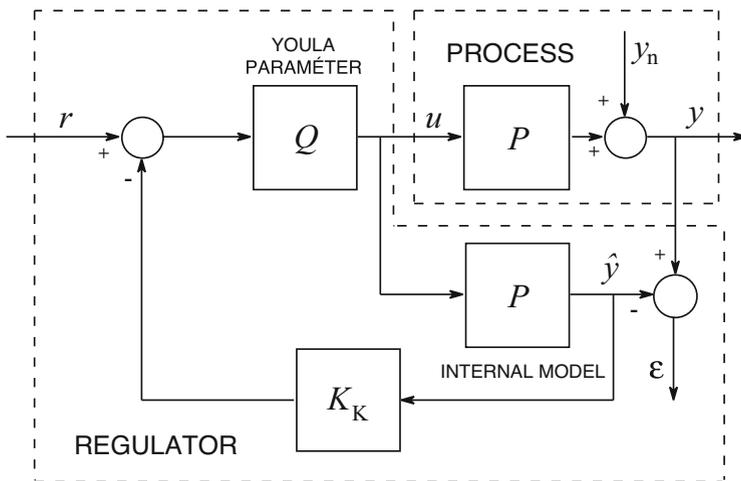


Fig. 11 Combination of the state-feedback principle and the IMC regulator

Fig. 12 The equivalent scheme of the state-feedback and the IMC regulator

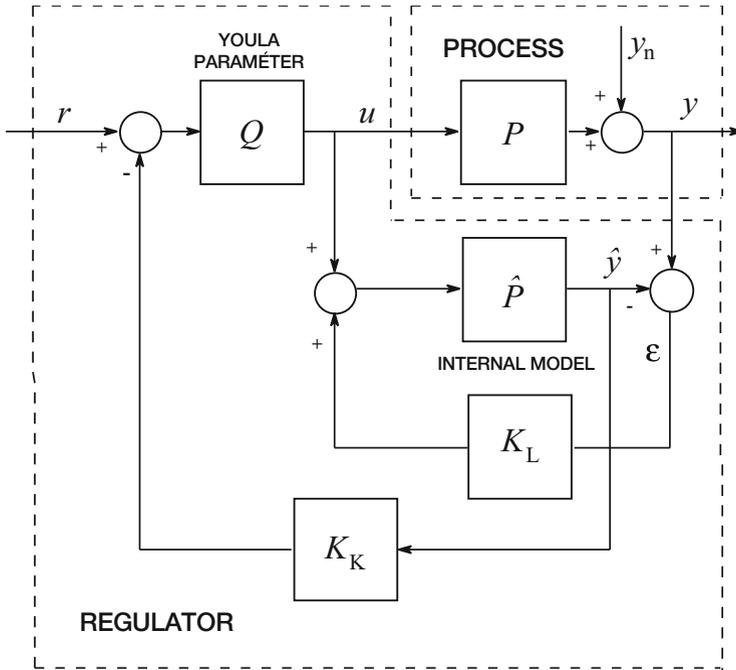
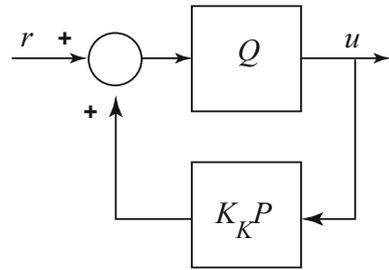


Fig. 13 Combination of the observer principle and the IMC regulator

In case of model error, the observer principle can be applied and the scheme of Fig. 13 is obtained. The model error now is

$$\varepsilon = y - \hat{y} = \frac{\varepsilon_o}{1 + K_L \hat{P}}; \quad \varepsilon_o = y - \hat{y}_o = Pu - \hat{P}u = (P - \hat{P})u \quad (23)$$

where

$$\hat{y} = \hat{P} \frac{1 + K_L P}{1 + K_L \hat{P}} \Big|_{\hat{P}=P} u = Pu \quad (24)$$

The process input and outputs are now

$$\begin{aligned} u &= \frac{Q(1+K_L\hat{P})}{1+\hat{P}(K_L+QK_K)+QK_KK_L\hat{P}P} \Bigg|_{\hat{P}=P} & r &= \frac{Q}{1+QK_KP}r \\ y &= \frac{QP(1-K_L\hat{P})}{1+\hat{P}(K_L+QK_K)+QK_KK_L\hat{P}P} \Bigg|_{\hat{P}=P} & r &= \frac{QP}{1+QK_KP}r \end{aligned} \tag{25}$$

what prove, if the internal observer-loop is designed well, providing very small ε , then the overall transfer functions will be independent of K_L . For example if very high gain K_L is selected, then

$$u = \frac{Q(1+K_L\hat{P})}{1+\hat{P}(K_L+QK_K)+QK_KK_L\hat{P}P} \Bigg|_{K_L \rightarrow \infty} \quad r = \frac{Q}{1+QK_KP}r \tag{26}$$

and the effect is the same when the model equals to the process (see (25)), i.e., $\hat{P} = P$.

Theoretically, we can obtain the simple IMC case (13) back, independently of K_K if

$$K_L = P^{-1} \tag{27}$$

is used in the internal observer loop [1, 2, 9]. The other system variables are

$$\begin{aligned} \varepsilon &= y - \hat{y} = Pu = y; & \hat{y} &= 0 \\ u &= Qr \\ y &= QPr \end{aligned} \tag{28}$$

The practical realization of the optimal combined scheme is shown in Fig. 14, where \hat{P} is used instead of P in (26) and in Fig. 13. Please note that \hat{P}^{-1} is usually a high-pass filter.

The model error and the process input and outputs are

$$\varepsilon = y - \hat{y} = \frac{\hat{P}}{\hat{P} - 1} \varepsilon_o \tag{29}$$

$$\begin{aligned} u &= \frac{Q}{1+QK_K\frac{P-\hat{P}^2}{1-\hat{P}}} \Bigg|_{\hat{P}=P} & r &= \frac{Q}{1+QK_KP} \Bigg|_{K_K \rightarrow 0} r = Qr \\ y &= \frac{QP}{1+QK_K\frac{P-\hat{P}^2}{1-\hat{P}}} \Bigg|_{\hat{P}=P} & r &= \frac{QP}{1+QK_KP} \Bigg|_{K_K \rightarrow 0} r = QPr \end{aligned} \tag{30}$$

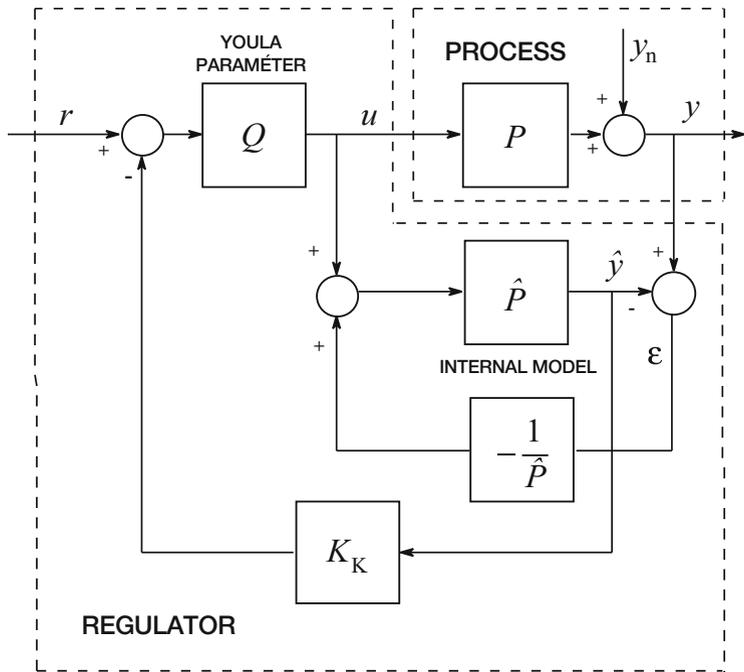


Fig. 14 The optimal combined scheme

So, in the other loop, the $K_K \rightarrow 0$ condition provides the original transfer functions. As a conclusion, the observer loop must be tuned for a high-pass filter and the feedback loop must be a low-pass filter. The necessary conditions can be reached only in the high-frequency and low-frequency domains, respectively.

6 Simulation Example

6.1 Example 1

The continuous process to be controlled is given by the transfer function

$$P(s) = \frac{1}{1 + 10s} e^{-30s} \tag{31}$$

The process is sampled with sampling period $T_s = 5$ s. At the input, zero-order hold is applied. Discrete control is realized. The sampled, digitalized output signal is forwarded to the process control computer via A/D converter, and the computer

calculates the control signal in each sampling point with a discrete algorithm and forwards it to the process input via a D/A converter.

Let us design a Youla-parameterized controller and a KB-parameterized controller and compare their behavior for reference signal tracking and output disturbance rejection.

The pulse transfer function is

$$G(z) = \frac{0.3935}{z - 0.6065} z^{-6} = G_+(z) z^{-6} \tag{32}$$

6.2 Youla-Parameterized Controller Design

The reference signal filters according to Fig. 8 are given by transfer functions

$$R_n(s) = \frac{1}{1 + 5s} \quad \text{and} \quad R_r(s) = \frac{1}{1 + 8s} \tag{33}$$

and their corresponding pulse transfer functions are

$$R_n(z) = \frac{0.6321}{z - 0.3679} \quad \text{and} \quad R_r(z) = \frac{0.4647}{z - 0.5353}. \tag{34}$$

The best control is expected if the inverse model of the plant appears in the forward path of Fig. 8. But as the dead time cannot be inverted, only $G_+^{-1}(z)$ will appear in the Youla parameter.

$$Q = R_n G_+^{-1} = \frac{0.6321}{z - 0.3679} \frac{z - 0.6065}{0.3935} \tag{35}$$

6.3 KB Controller Design

According to Fig. 7 in the controller C , the parameter $Q = Q_1$ is chosen as the invertible part of the pulse transfer function of the plant.

$$Q_1 = G_+^{-1} / z = \frac{z - 0.6065}{0.3935z} \tag{36}$$

and the serial controller is

$$C = \frac{Q_1}{1 - Q_1 \cdot G} = \frac{2.541z^7 - 1.541z^6}{z^7 - 1} \tag{37}$$

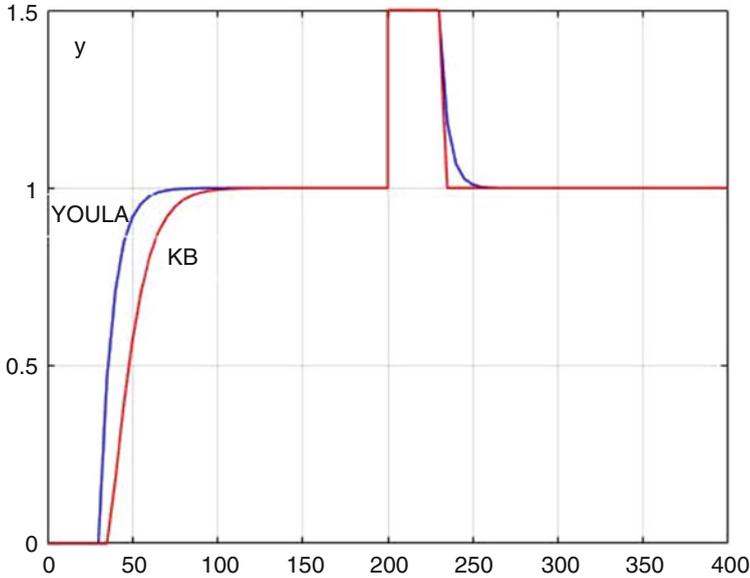


Fig. 15 Output signals of the Youla- and of the KB-parameterized control systems

Q_1 and so C will influence the disturbance rejection, while Q_r , which is chosen equal to R_r in the Youla-parameterized structure, influence the reference signal tracking.

Figure 15 shows the output signals in the two circuits. The reference signal is a unit step starting at $t = 0$ s, and the step disturbance with amplitude 0.5 acts at $t = 200$ s.

Both controllers give a nice control performance. The difference is due to the fact that in the Youla-parameterized controller, two filters influence the control behavior, while in the KB structure, Q_r influences the tracking performance, and Q_1 separately influences the disturbance rejection.

7 Conclusions

It is shown that the different advanced control-loop parameterization methods have very interesting relationships. If we use model-based forms, then these connections are even more sophisticated.

This chapter tries to clarify these relationships from the Youla and KB parameterization via the state feedback topologies and observer principle.

References

1. K.J. Åström, B. Wittenmark, *Computer Controlled Systems* (Prentice-Hall, 1984), p. 430
2. I.M. Horowitz, *Synthesis of Feedback Systems* (Academic Press, New York, 1963)
3. L. Keviczky: Combined identification and control: another way. (Invited plenary paper.) in 5th IFAC Symp. on Adaptive Control and Signal Processing, ACASP'95, 13–30, Budapest, Hungary (1995)
4. L. Keviczky, Cs. Bányász, Optimality of two-degree of freedom controllers in H_2 - and H_∞ -norm space, their robustness and minimal sensitivity, in 14th IFAC World Congress, F, 331–336, Beijing, PRC (1999)
5. L. Keviczky, C. Bányász, Iterative identification and control design using K-B parameterization, in *Control of Complex Systems*, ed. by K. J. Åström, P. Albertos, M. Blanke, A. Isidori, W. Schaufelberger, R. Sanz, (Springer, 2001), pp. 101–121
6. L. Keviczky, C. Bányász, *Two-Degree-of-Freedom Control Systems (The Youla Parameterization Approach)* (Elsevier, Academic Press, 2015)
7. L. Keviczky, R. Bars, J. Hetthéssy, C. Bányász, *Control Engineering* (Springer, 2018)
8. L. Keviczky, R. Bars, J. Hetthéssy, C. Bányász, *Control Engineering: MATLAB Exercises* (Springer, 2018)
9. J.M. Maciejowski, *Multivariable Feedback Design* (Addison Wesley, 1989), p. 424

A Virtual Serious Game for Nursing Education



Youxin Luo, Abel A. Reyes, Parashar Dhakal, Manisa Baker, Julia Rogers, and Xiaoli Yang

1 Introduction

Healthcare professionals are required to follow standard procedures during patient care. Deviation from standards leads to serious adverse events and complications. Therefore, to ensure the safety of patients, professionals must learn the standard procedure and practice it repeatedly before they perform care to real patients. Traditionally, nursing students were taught standard procedure skills in class and were required to demonstrate theoretical competency in the simulation laboratory. Due to limitation of lab hours, deficiency of equipment, decline in nurse educators, and financial implications, the opportunity for students to practice the standard procedures has been restricted [1, 2]. Additionally, faculty face challenges of large class sizes, space constraints, and limited manikins affecting the value of the instruction provided and student retention. Therefore, students are not able to receive adequate feedback and assistance to be competent in vital skills. Recent research in the healthcare field [3, 4] have demonstrated that traditional methods of learning are inadequate in developing competency and proficiency partly due to constraints of the simulation laboratory. A safe and efficient supplementary tool for practicing procedure and problem-solving skills for nursing students is needed.

Y. Luo (✉) · A. A. Reyes · X. Yang

Electrical and Computer Engineering, Purdue University Northwest, Hammond, IN, USA
e-mail: luo300@pnw.edu; areyesan@pnw.edu; yangx@pnw.edu

P. Dhakal

Electrical and Computer Science Engineering, The University of Toledo, Toledo, OH, USA
e-mail: Parashar.dhakal@utoledo.edu

M. Baker · J. Rogers

College of Nursing, Purdue University Northwest, Hammond, IN, USA
e-mail: baker417@pnw.edu; jlrogers@pnw.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_68

927

Researchers have found the use of an interactive skill development-based video game as a promising teaching methodology in improving standard procedure skills [5]. Interactive games with sufficient complexity and playability have shown to efficiently improve skills among nursing students as compared to the traditional teaching methods. Moreover, the interactive environment provided in the game helps enhance the retention of knowledge, increase student engagement in learning, promote problem-based active learning, and encourage critical thinking [6]. Additionally, a recent survey by researchers in the field shows that many of the nursing students were supportive and had a positive attitude toward the use of new media technologies as a complement to traditional nursing education [7]. There are a multitude of supplemental tools currently available that are presented as 3D book-based tutorials; however, they lack intractable objects and corresponding feedback that provides a vivid impression for memory retention and enough stimulation for critical thinking [6]. Nurses must incorporate critical thinking skills into the performance of a procedural skill for patient safety; therefore memorization alone does not promote skill mastery. The incorporation of critical thinking increases the educational value. The interactive environment should be very close to real-world environment to assist students in transitioning from training to real-life practicing. Maintaining a domain as close to a real experience as possible provides engaging, immersive, and authentic context of learning as compared to traditional virtual environment.

With this motivation, we designed a 3D virtual hospital environment with multiple interactive models, real-world pictures, tutorial videos, user-friendly graphical interface, and an evaluation system. This 3D design allows nursing students to practice standard procedure skills. The design helps students in simulating the procedure of insertion of nasogastric tube, allowing the user to practice the procedure step by step in a third-person perspective, simulating corresponding behaviors such as checking of patient's wrist band, raising bed, placing supplies, and updating score and report. The game is designed for single user and is Windows based.

2 Background and Related Work

According to the research [8], currently healthcare is experiencing a nursing shortage exacerbated by a nursing faculty shortage and increasing workloads for faculty. One solution to these challenges would be the use of simulation tool and distance education. Serious game has been confirmed to be useful and cheaper training technology for health-related education, and many researchers have been focusing on this area of research through design and implementation of such games [9]. Researchers have designed video-based scenarios with real chronic obstructive pulmonary disease (COPD) patients and a registered nurse as an actor to build a game in order to aid nursing students learn clinical reasoning and decision-making skills [10]. The reviews on such games included comments of being useful,

usable, and satisfying from the end users. In addition, games like *Doc and the MACHINE* provided real-time assessment and feedback in cultural competence training. It also provided the learners with answers to specific questions or problems [11]. Games proposed in [12] simulated different scenarios that usually happen in emergency rooms to familiarize students with emergency room atmosphere. Similarly, as according to the researchers in [13], *PHARMACOMM*, an interactive tool for pharmacy education, updates the medications as response to interaction with patients.

Although there are already many serious games designed for nursing education in the market, most games only cover one or two aspects of the core competencies (specific procedural, health assessment, or communication skills) [14]. Therefore, there is a need for more educational games focused on the mastery of fundamental skills, tasks, activities, and operations to match the desired learning outcomes [14, 15]. In the paper, we describe a new serious game which combines 3D model, real tutorial videos, and real pictures to make the environment look more realistic. In addition, the game consists of a user-friendly graphical interface to drive the story and help students retain knowledge. It features animations to stimulate engagement, active learning, and critical thinking. Moreover, the proposed game incorporates the user's performance and assesses the student's progress. This enables the student evaluate their performance in real time and increase critical thinking skills.

3 Game Design

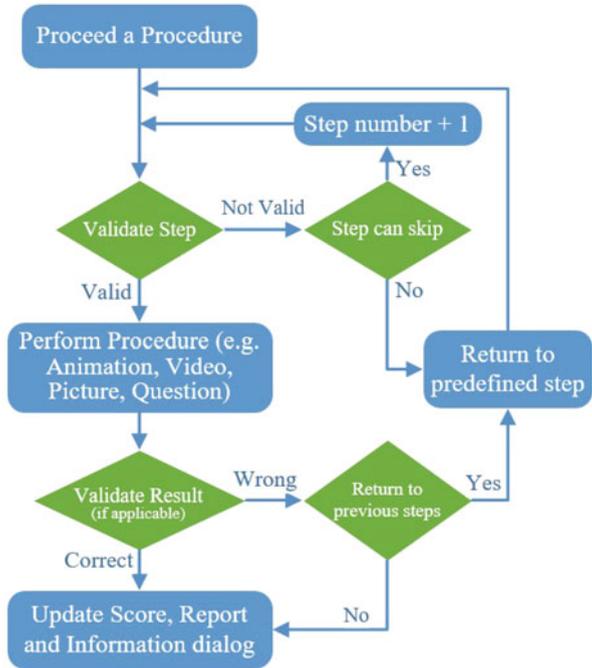
In our game, in order to implement all the desired features such as pictures, videos, 3D environment, animations, and user-friendly graphical interface and to reserve all the potential platform such as mobile, virtual reality (VR), and augmented reality (AR), we chose Unity game engine as software development platform [16].

The game starts with a menu featured with two options, either to directly start practicing of procedure skills or a tutorial video option to watch the standard procedure for the first-time users. In the tutorial video, the user has options to either go back to the menu or start practicing instead of having the whole tutorial completed. Once the practice begins, the game displays a story as depicted in Fig. 1 of a sick person who needs the insertion of nasogastric tube by the user. Then, the user clicks on the "Challenge Accepted" button to start the journey; the game is story based to attract user's attention and stimulate active learning as it is one of our primary goals. One challenge former research [17] faced in such story based game is whether to let user choose the wrong answer or not during the procedure training process. Some serious game allows the user to choose wrong answers [18], while others do not [19]. To solve this challenge, we proposed a new strategy which allows the user to select wrong answer; however, instead of ending the game, the game would redirect the user to a former step where the wrong procedure would cause problem. Using this method of redirection allows the user to continue in the game and assist the user to identify the error in the step. This ultimately provides

Fig. 1 Practice start: Welcome page with backstory



Fig. 2 Game mechanism flowchart



the user with direction on where they can safely restart the skill. The workflow of the mechanism is shown in Fig. 2, where each step the user chooses to proceed will trigger the decision tree and gives responding result. The detailed mechanism on this would be discussed in Section 4. Finally, when the user reaches the last procedure, the report would automatically be generated and presented for user to evaluate their performance.

4 Game Implementation

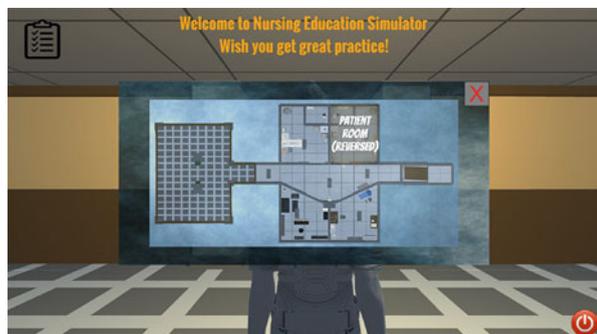
Serious gaming enhances patient safety, standardizes training, lowers cost, increases accessibility, and can be adjusted to meet specific learning objectives [20]. However, if students do not engage and partake in the serious game objective to learn new skills, it will not be effective [21]. Therefore, like general video games, it has to be made fun in order to encourage its usage and to increase memory retention through practical learning. To achieve aforementioned goals, the game design should involve different factors such as backstory, realism, adaptability, interaction, and feedback [22]. The proposed game begins with a backstory to provide a setting and guide the user through the game. In order to provide a realistic setting for the game, we designed a 3D hospital environment that replicates a real hospital environment. Further, in order to make the game playable, we incorporated interactions and animations to 3D models such as computer, supply closet, multifunctional hospital bed, and many more. Moreover, a novel game mechanism was created to address wrong step selection by the user. Finally, we implemented a real-time score and report system to evaluate user's performance and to give feedback. More detailed implementation would be presented in the following subsections.

4.1 3D Hospital Environment

The 3D hospital environment is the first stage to ensure user experiences a better practice by easily transiting their procedure skills from a serious game to real-life operation. Therefore, to ensure realism, the 3D hospital environment was designed based on pictures of real hospital.

The top view map of the hospital is shown in Fig. 3; it includes a waiting room, two patient rooms, a storage room, office room, and a long aisle. The waiting room depicted in Fig. 4a is the start point of the game located at the left of the map. When the practice starts, the user would be informed with the backstory and would see the main character as a third person and then starts the journey by getting through the

Fig. 3 QuickAccess panel



door ahead. Figure 4b shown is the view of the long aisle, which is the center line of the hospital; it connects to all the rooms with door and stores items such as personal protective equipment (PPE) box, hand sanitizer, sink, and waste bin.

Patient room (Fig. 4c) is the main room that was developed for the procedure insertion of nasogastric tube, as shown in Fig. 5. The room has a patient lying on the multifunctional hospital bed in the middle, with all the medical equipment around, table and chairs for visitors, and a well-designed rest room behind the wall in the back. The second patient and office room are reserved for more procedures and both have related models set up. Storage room (Fig. 4d) is used for checking medical records and storing of medical equipment and supplies, computer, supply closet, and all other kinds of equipment.

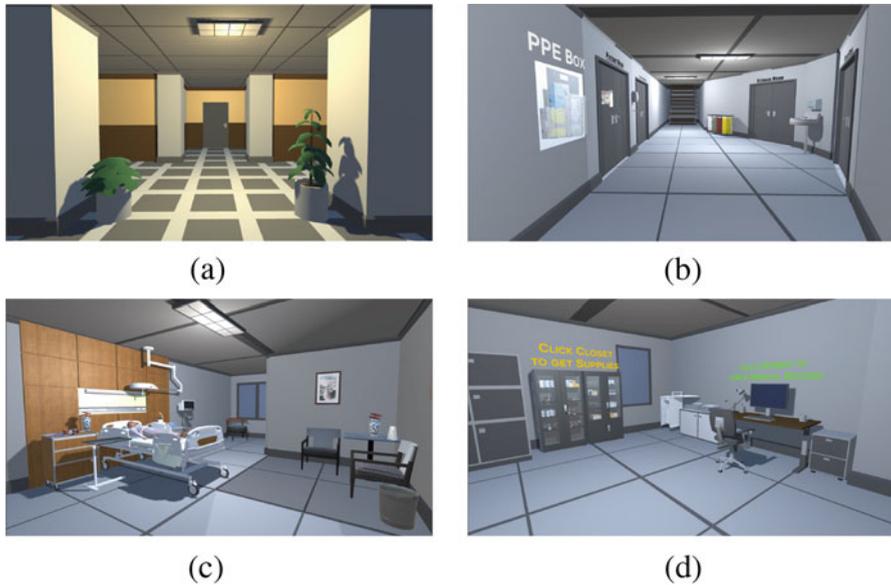


Fig. 4 3D Hospital environment. (a) Waiting room. (b) Aisle. (c) Patient room. (d) Storage room

Fig. 5 Patient and medical equipment



All the 3D models for our game were downloaded from 3D model websites [23–25] or imported from Unity assets store [26] and have been modified in 3ds Max [27] or rearranged in Unity [16] to build the environment. In addition, all these models are reusable for other procedures necessary within nursing and can be easily modified to meet future needs.

4.2 Interaction and Animation

Featured interaction and animation help reduce cognitive load of learning, and reduction of cognitive load can result in more effective learning [28]. And to achieve such effective learning, we designed many interactions between the user and the game along with multiple intriguing animations to make the game playable and fun for the user.

Interactive Objects To separate interactive objects from others in the proposed game, the cursor of mouse has been set to change icon when it hovers over interactive objects. Beside the interactive 3D model objects, all the graphical user interfaces have interaction with the user, and this will be discussed in the “Graphical User Interface” section. The interactive objects in the game have been listed below with corresponding actions:

- All the doors: Click door to open/close.
- Computer screen: Click on the screen to start computer, and then click the record to open medical record panel with detailed information on medical order and record.
- Supply closet: Click to open the supply panel and inventory panel to fetch supplies.
- Clipboard and precaution board: Click to bring up the board for information of patient.
- Hand sanitizer/sink: Click to perform hand hygiene.
- Personal protective equipment (PPE) box: Click PPE to put on items such as gloves, gown, and mask.
- Medical table: Click to move table to bedside.
- Handrail of the hospital bed: Click to raise/lower the handrail of the bed.
- Waste bin: Click to open/close the lid.
- Trash can: Click to perform remove extra equipment.

Animations This game features multiple animations for objects to provide an intriguing environment for the user to provide a fun experience while practicing skills. It incorporates critical thinking and decreased the memorization of a concept or step. Many objects in our design have been animated to achieve the goal such as:

- All the doors: Animated to open and close.
- Waste bin: Animated to open/close the lid.
- Computer screen: Animated to power on and off with gradient animation.
- Hand sanitizer and PPE box: Animated to change color while user hovers over it.
- Medical shadowless lamp: Animated to stretch, rotate, light up, and reverse.
- Medical table: Animated to change height, move to bedside, and place supplies as shown in Fig. 6.
- Multifunctional hospital bed: Animated to raise/lower bed, change fowler to 30/60/90°, remove pillow when fowler reaches 90° degree, and lower/raise the handrail. Figure 6 shows the animated raised bed with 90° high fowler and lowered handrail.
- Patient: Animated to have small motion while idle, raise hand and rotate head for wrist band checking, and change body posture according to fowler's angle as shown in Fig. 6.
- Graphical user interface: Animated to change color, size, position, etc. for interaction.

4.3 Graphical User Interface (GUI)

GUI is an important part in game design; it conveys information, provides human-computer interaction, and enhances the efficiency of the underlying program logic. For our game, the target was to practice procedure skills by conveying nursing knowledge through explanation, interaction, and evaluation to the user. To efficiently convey the information and to make the game easy to use, several user-friendly graphical interfaces were designed.

Runtime Interface As shown in Fig. 6, when the practicing of procedure is activated, the runtime interface has the following features: a “Menu” button on the top-left corner, an “Information dialog” on the top, a “Score” display on the top-right corner, and an “Exit” button on the bottom-right corner:

Fig. 6 Multifunctional hospital bed



- The “Menu” button opens the main menu panel and has five options for checking the PPE worn, opening control panel, checking report, opening checklist, and using “QuickAccess.”
- “Information dialog” displays the information of procedure tracking such as step validation, step completed, and the error made when the procedure performed is wrong.
- “Score” keeps track of the earned score during the current procedure practice and provides feedback to the user in real time.
- The “Exit” button provides the option of quitting game at any point of time in the game.

Three-Level Control Panel Figure 7 presents the three-level control panel, which controls most of the interactive features in the game. The first level is “Main Menu” which has five options for general functions; the second level is “Control Panel” which lists all the procedure categories; and the third level is “Procedure Panel” where all the individual procedures are located:

- “Main Menu” as shown in Fig. 7a can be accessed by clicking the “Menu” button as mentioned in section “Runtime Interface.” It has five general functions: “PPE” for opening PPE-worn panel to check and take off equipped PPE; “Control Panel” for opening the second-level panel; “Report” for opening report panel to allow user to track progress and score; “CheckList” to provide the standard procedure

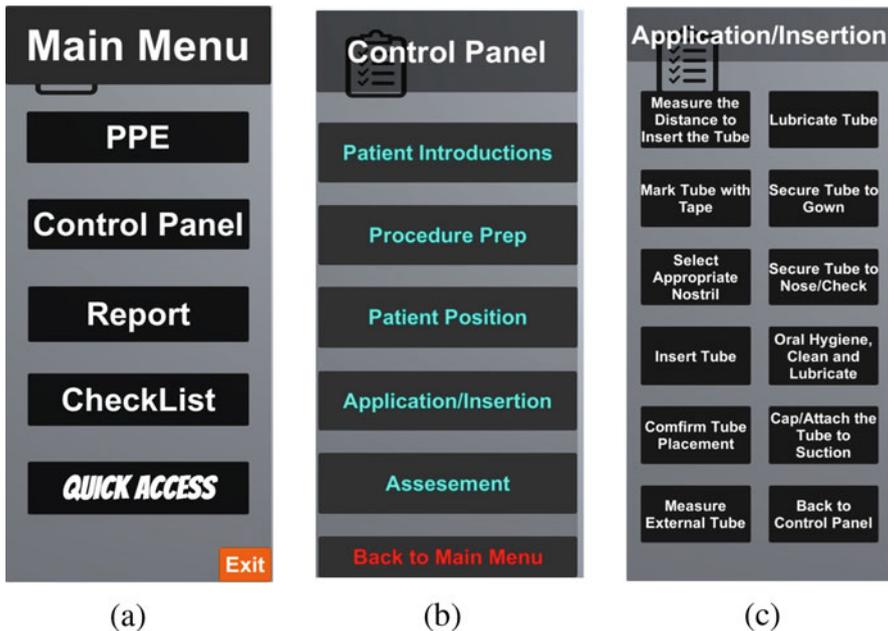


Fig. 7 Three-level control panel. (a) Main menu. (b) Control panel. (c) Procedure panel

checklist to user as a reference to complete the practice; and “QuickAccess” as shown in Fig. 3 serves as a quick transit method to transfer game character from one location to another making the process convenient to the first-time users and saving their time from doing unnecessary moves as well.

- To help user locate specific procedure, “Control Panel” was designed to categorize all the procedures into five different categories as shown in Fig. 7b. In this way, user can easily find the procedure they want to proceed instead of searching a procedure in a table that contains all the procedures which can usually be more than 25.
- Level 3 is “Procedure Panel,” and it has five different panels where each panel represents one category of procedures. Figure 7c depicts one of the procedure panels which contains up to 11 individual procedures with an option of “Back to Control Panel” as well. User can proceed a procedure by clicking the procedure button, and the game would provide corresponding actions and feedback to user accordingly.
- To make the interface user-friendly, in level 3, the procedure panels have an option on the bottom right to go back to the second level—control panel. Moreover, the second level panel also has an option to return to the first level on the bottom as shown in Fig. 7b, and the first level can be closed by clicking the exit button on the bottom right of the main menu. These three options allow the user to easily switch between three-level control panel and the hospital environment.

Supply and Inventory Gathering of needed supplies before performing care to patient is critical. To simulate this procedure, we designed a supply-inventory system as shown in Fig. 8. The system simulates user collecting supplies from supply closet and putting into inventory as depicted in Fig. 8a:

- A supply closet panel was designed to display all the supplies in the store with a picture and name indicated on each supply. User can select supply by clicking the picture in supply panel, and it would be stored in the inventory.
- Inventory panel stores all the items picked from supply with picture and name. A “Remove” button has also been provided for removing unexpected supply from the inventory. Moreover, the inventory panel can also be hidden/shown by clicking the arrow on the top. Inventory panel can be accessed at any time after the supply closet is opened and user can use supplies in inventory to perform procedure by clicking the supply icon as shown in Fig. 8b.

Multiple Learning Panels To enhance the complexity and playability and to evaluate application of concepts, multiple learning panels as shown in Fig. 9 were implemented. Unlike traditional explanation-based user interface (UI), we designed several panels using different learning methods and materials, namely, multi-choice, true or false, input-field, and video-player to help user enhance knowledge. Detail regarding the panel used has been elaborated below:

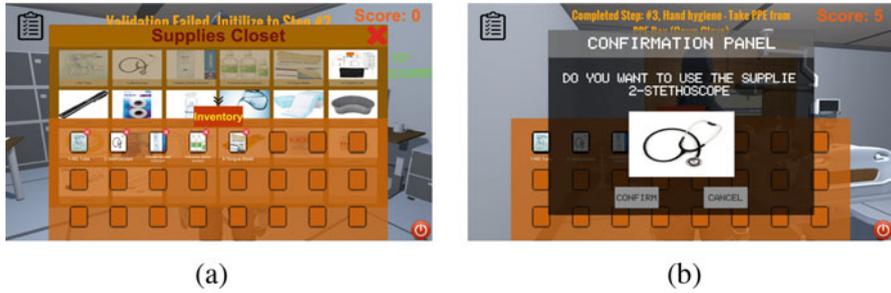


Fig. 8 Supply-inventory system. (a) Supply and inventory panel. (b) Using supply in inventory

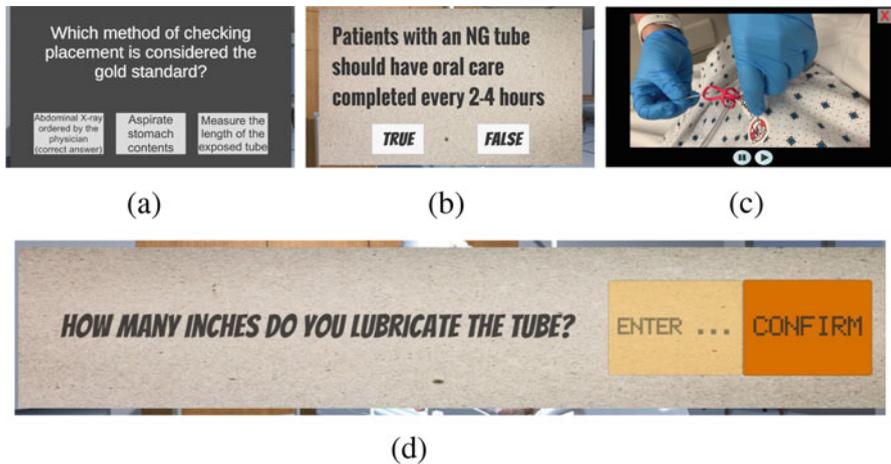


Fig. 9 User interface components. (a) Multi-choice. (b) True or false. (c) Video-player. (d) Input-field panel

- Figure 9a presents the multi-choice panel, which allows the user to choose one or more correct answers from the choice listed. Moreover, it also gives user flexibility to replace the text information with picture.
- Figure 9b shows the true or false panel; in this panel users are questioned in such a way that they have to just choose either true or false for any particular question. This approach has advantage over multi-choice panel because user does not have to read choices and it saves time.
- Figure 9c depicts video-player panel; it provides video tutorial instead of giving text information to help understand complex procedures.
- Figure 9d presents the input-field panel that allows user to give their own answer regarding the question to trigger and enhance knowledge.

To ensure the user can start the game and effectively practice procedure skills, helpful settings were implemented. This involved setting of panel to be transparent

to allow user to gain a better perspective. Moreover, it also gives the user flexibility to change view by simply dragging the mouse. Further, to avoid user from accident interactions with the model, the game was set to prevent changing the view when clicking through UI. In addition to this, double confirmation was also implemented to interactive objects and some UI components to further avoid accident interaction.

4.4 Evaluation System

In this section, we propose an evaluation system to record procedures performed by the user and to provide user corresponding feedback. The basic mechanism of the evaluation system has been shown in Algorithm 5, and the final report generated by the evaluation system has been depicted in Fig. 10. As shown in Fig. 10, the report consists of two parts: procedure step record that records performed procedure information and score that provides credit point for each correct procedure.

Algorithm 5 Procedure evaluation system

```

1: for Every procedure step do
2:   if This step is valid to proceed then
3:     Perform step procedure
4:     if Question not required or answer is correct then
5:       Update Record, Score, report, and Information Dialog
6:     else
7:        $i \leftarrow$  The predefined return step number for this step
8:       while  $i \leq$  CurrentStepNumber do
9:         Initiate game setting, Record, Score and Report for step  $i$ 
10:         $i = i + 1$ 
11:       end while
12:     end if
13:   else
14:     Initiate game from predefined return step to current step
15:   end if
16: end for

```

The term procedure step means step number, which has been used as validation criteria in “validate step” of the flowchart as shown in Fig. 2, and the actual procedure performed in game is interaction and question. The procedure step and score both start at 0, and each procedure the user performs should pass the step validation; then the right answer about the procedure question (if applicable) given by the user will be counted as credit. After that, the evaluation system records the information of the completed procedure and updates the score, report, and information dialog as illustrated in Fig. 6 in the real time.

The step validation process works as shown in Algorithm 5. To start with, normally a step procedure is valid if it is the next step of the last procedure completed. However, there are some exceptions such as the procedure “explain



Fig. 10 Evaluation system. (a) Report: top. (b) Report: bottom

discomfort to the patient” which would be a good practice but not a necessary procedure to proceed successfully. Therefore, procedures that would not affect the fundamental functionality would be categorized as “can be skip” in our application as shown in Fig. 2 and would only cause losing out some points without affecting to continue the procedure. However, other procedures that are much critical and might affect former procedures such as “Insert Tube” would cause an initiating back to a predefined step to eliminate all the possible problems. Besides, if the procedure proceeds with a question, then the user needs to provide a correct answer to gain credit, and incorrect answer would be treated same as an invalid step.

Therefore, if the procedure is not valid or an incorrect answer for procedure is given, the system decides which former step the game needs to initiate from based on the predefined standard procedure provided by the healthcare professionals in the field. In this case, the game starts from the predefined procedure step because of the wrong step/answer given, then it clears all the record and score till the current step. The information dialog on the top would inform the user the reason for the failure of step validation, or the answer provided was not correct and further provide information on which step the user should start from.

By following the standard procedure and the guide, when the end of the game is reached, the user is guaranteed to practice the standard procedure completely at least one time. Moreover, the system also keeps track of the procedures performed by the user and updates record and score after a complete procedure step is finished. User can access the report anytime during the game through the main menu panel, and when the final procedure is completed, the report is automatically generated and presented to the user as a feedback. The determination of fairness will be determined when evaluated in the future work.

5 Conclusion and Future Work

We implemented an innovative serious game as a complementary tool for undergraduate nursing students to practice procedure skills. In the game, we created a realistic

3D hospital environment, featured plenty of intriguing interaction and animation, designed different kinds of user-friendly graphical interface, and developed a novel evaluation system. The interaction and animation in our game provide fun and playability to user for sustained learning. The various GUI and the evaluation system bring considerable complexity. The realistic hospital environment makes the game an effective supplement for nursing education. Finally, the proposed game allows user to practice procedure skills in a safe and effective environment and helps solve the problem of increasing enrollment, deficiency of nurse educators, and insufficient simulation opportunities in the nursing field.

In regard to the future work, there are two directions that can help improve the current result. One direction is to design a comprehensive evaluation plan to assess the effectiveness of the game, and the other direction is to implement more procedure for practicing purpose. Further, this game can also be transferred to other portable platforms such as mobile devices and VR/AR.

References

1. A.L. Butt, S. Kardong-Edgren, A. Ellertson, Using game-based virtual reality with haptics for skill acquisition. *Clin. Simul. Nurs.* **16**, 25–32 (2018)
2. M. Verkuyl, L. Atack, P. Mastrilli, D. Romaniuk, Virtual gaming to develop students' pediatric nursing skills: a usability test. *Nurse Educ. Today* **46**, 81–85 (2016)
3. L. Gonzalez, M.L. Sole, Urinary catheterization skills: one simulated checkoff is not enough. *Clin. Simul. Nurs.* **10**(9), 455–460 (2014)
4. S. Kardong-Edgren, P. M. Mulcock, Angoff method of setting cut scores for high-stakes testing: Foley catheter checkoff as an exemplar. *Nurse Educator* **41**(2), 80–82 (2016)
5. T.B. de Araujo, F.R. Silveira, D.L.S. Souza, Y.T.M. Strey, C.D. Flores, R.S. Webster, Impact of video game genre on surgical skills development: a feasibility study. *J. Surg. Res.* **201**(1), 235–243 (2016)
6. M.A. Royse, S.E. Newton, How gaming is used as an innovative strategy for nursing education. *Nurs. Educ. Perspect.* **28**(5), 263–267 (2007)
7. J. Lynch-Sauer, T.M. VandenBosch, F. Kron, C.L. Gjerde, N. Arato, A. Sen, M.D. Fetzters, Nursing students' attitudes toward video games and related new media technologies. *J. Nurs. Educ.* **50**(9), 513–523 (2011)
8. C. Fitzgerald, I. Kantowitz-Gordon, J. Katz, A. Hirsch, Advanced practice nursing education: challenges and strategies. *Nurs. Res. Pract.* **2012**, 854918 (2012)
9. F. Ricciardi, L.T. De Paolis, A comprehensive review of serious games in health professions. *Int. J. Comput. Games Tech.* **2014**, 787968 (2014)
10. H.M. Johnsen, M. Fossum, P. Vivekananda-Schmidt, A. Fruhling, Å. Slettebø, Teaching clinical reasoning and decision-making skills to nursing students: design, development, and usability evaluation of a serious game. *Int. J. Med. Inf.* **94**, 39–48 (2016)
11. S. Mayr, S. Schneider, L. Ledit, S. Bock, D. Zahradnicek, S. Prochaka, Game-based cultural competence training in healthcare, in *2017 IEEE 5th International Conference on Serious Games and Applications for Health (SeGAH)* (2017), pp. 1–5
12. E. Lotfi, A. Belahbib, A digital revolution in nursing education—the serious games, in *2016 5th International Conference on Multimedia Computing and Systems (ICMCS)* (2016), pp. 705–709

13. R.T. Lambertsen, S. Tang, J. Davies, C. Morecroft, Serious gaming for pharmacy education: development of a serious games for teaching pharmacist communication and drug administration in a virtual hospital setting, in *2016 9th International Conference on Developments in eSystems Engineering (DeSE)* (2016), pp. 151–156
14. A.J.Q. Tan, C.C.S. Lau, S.Y. Liaw, Paper title: Serious games in nursing education: an integrative review, in *2017 9th international conference on virtual worlds and games for serious applications (VS-Games)* (IEEE, Piscataway, 2017), pp. 187–188
15. T.M. Connolly, E.A. Boyle, E. MacArthur, T. Hainey, J.M. Boyle, A systematic literature review of empirical evidence on computer games and serious games. *Comput. Educ.* **59**(2), 661–686 (2012)
16. Unity, Unity real-time development platform 3d, 2d vr & ar. Accessed 22 March 2020. <https://unity.com/>
17. H.M. Johnsen, M. Fossum, P. Vivekananda-Schmidt, A. Fruhling, Å. Slettebø, Developing a serious game for nurse education. *J. Gerontol. Nurs.* **44**(1), 15–19 (2018)
18. J. Kaczmarczyk, R. Davidson, D. Bryden, S. Haselden, P. Vivekananda-Schmidt, Learning decision making through serious games. *Clin. Teacher* **13**, 07 (2015)
19. F. Laamarti, M. Eid, A. El Saddik, An overview of serious games. *Int. J. Comput. Games Tech.* **2014**, 10 (2014)
20. R. Wang, S. Demaria, A. Goldberg, D. Katz, A systematic review of serious games in training health care professionals. *Simul. Healthcare* **11**, 41–51 (2015)
21. P. Backlund, M. Hendrix, Educational games—are they worth the effort? A literature survey of the effectiveness of serious games, in *2013 5th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)* (2013), pp. 1–8
22. W.S. Ravyse, A.S. Blignaut, V. Leendertz, A. Woolner, Success factors for serious games to enhance learning: a systematic review. *Virtual Reality* **21**(1), 31–58 (2017)
23. Renderpeople. Accessed 22 March 2020. <https://renderpeople.com/free-3d-people/>
24. 3dtk. Accessed 22 March 2020. <http://www.3dtk.com/2019/0717/36352.html>
25. Evermotion. Accessed 22 March 2020. <https://evermotion.org>
26. Unity asset store Accessed 22 March 2020. <https://assetstore.unity.com/>
27. Autodesk 3ds max, Accessed 22 March 2020. <https://www.autodesk.com/products/3ds-max/overview>
28. L. Gonzalez, S. Kardong-Edgren, Deliberate practice for mastery learning in nursing. *Clin. Simul. Nurs.* **13**, 10–14 (2017)

Modeling Digital Business Strategy During Crisis



Sakir Yucel

1 Introduction

COVID-19 is a major health, economic, and social crisis in the modern age. We witnessed very tough times on our lives and livelihoods with many people being hospitalized and unfortunately many losing their lives. We have been all worried about our wellness and safety and concerned about if we ever live the same life. It is a big health issue. It is also an economic crisis. Many businesses closed and many jobs were lost, some temporarily and some not, and with huge job loss spike not seen in recent time.

All countries have had some measures, for example, asking citizens to stay home, curb travel, and maintain physical distance and lockdowns of non-essential businesses. It has impacted people, economy, business and work, policies, and all aspects of our lives globally and caused a change on the social and economic order which leads to a new normal. How exactly it will evolve remains to be seen as there is still big uncertainty and volatility.

Many uncertainties still exist, but it is certain that there are and there will be changes. We will see change in our lives, behavior, economy, society, business, regulation, and all as we are trying to understand and shape the new normal. It impacted all of our lives and also impacted the digital business strategies of corporations. Even before the COVID-19 pandemic, digitization had changed the consumer behavior and habits, regulations, supply-side factors, demand-side factors, and costs of information structure and coordination. We have been experiencing shifts from physical interactions to digital interactions and transitioning from physical, predictable, and slow world into digital, virtual, fast, and agile world. However,

S. Yucel (✉)
Wexford, PA, USA
e-mail: yucel@bluehen.udel.edu

the impact of the COVID-19 is beyond the disruption that digitization has caused. COVID-19 will likely accelerate the digitization but will also force corporations to refine and possibly redefine their digital business strategies.

As [1] defines the crisis as “a sequence of events that can have substantial negative consequences if not managed appropriately,” corporations could experience negative consequences if they cannot manage it appropriately but could also find some opportunities. Although each crisis is unique, corporations could learn common patterns, lessons, and managerial practices that they can program into their business strategy [1]. Corporations have to address many questions about how to deal with the crisis in digital age and refine their digital business strategy while going through some transformations already. The main question is how corporations could navigate through this crisis when traditional economy and even digital economy assumptions and approaches do not necessarily apply? How will the macro-economy look like? What should be the new digital business model during and after the crisis, how to define and how to transition to it? How quickly and on what scale will the crisis likely impact the industry and the business model of the corporation? Where will the most challenging threats come from? How big are the threats opportunities? What are the digital opportunities that came with the crisis? What are the best responses to embrace the opportunities and meanwhile protect against the threats? What digital transformation initiatives and strategic options are most effective? What are the benefits of the transformed business? What risks exist during the transformation and after the transformation?

In our previous work, we addressed the challenges with modeling digital business strategy and digital transformation strategy and estimating the benefits, drawbacks, and risk of digital transformation strategy [2–4]. In this paper, we extend our work by addressing how the corporations could characterize the digital business strategy during crisis and devise a framework for how they could model it and how they could evaluate various strategic options. Although some of the discussion will revolve around COVID-19 pandemic, we try to keep much of it generic to crisis in general. Making any prediction is highly challenging during COVID-19 pandemic, and we will not try that in this paper. We will try to devise a framework for organizations to model the digital business strategy during crisis in general.

Importance of digitization and digital transformation became clear during the crisis. Corporations that made progress in their digital transformations have been successful during the crisis. Digitization plays an even more significant role during the crisis in the success of the corporations. In the digital economy even without a crisis like the COVID-19 pandemic, companies had to observe and consider far more dynamics to understand the nature of the disruption [5]. Supply, demand, market structure, and competition are less constrained in the digital marketplaces. Digital disruption requires methodical evaluation, planning, and precaution. The corporations are challenged with understanding and evaluating the dynamics that play significant roles in the digital economy. In addition to all the challenges that come with digital economy, corporations now face the uncertainties and challenges of crisis. In this section, we will address how digital disruption during crisis relates to various other digital dynamics.

Digital disruption in crisis is characterized mainly by its threats, opportunities, digital marketplaces, macro level changes, and the stage of crisis. There are threats related to demand, supply, ecosystem, new market making, and new value proposition. There are additional threats due to digitization and technological threats from disruptive technologies. Similarly opportunities exist. Digital marketplaces are the digital platforms and ecosystems. Digital disruption and its impacts on industry and on corporation are additionally characterized by the current stage of the disruption and the current business model of the corporation, which is formed mainly in line with the current market conditions before the crisis. Corporations are interested in moving from current business model to transformed business model while navigating through the crisis. The vision for this transformation is in the digital business strategy. The digital business strategy is an organizational strategy that aims to create value by employing digital resources. Digital business strategies incorporate exploitation of business opportunities that arise from digital disruption. Digital business strategy should be developed in consideration of the strategic areas that the corporation is focusing on during the crisis and the stage the corporation is in handling the crisis. The transformed business model is derived from the digital business strategy. To implement the vision of the digital business strategy and to reach the transformed business model, digital transformation strategy is needed [6]. Digital transformation strategy focuses on corporation’s journey to achieve the desired future state of being digitally transformed [6]. Digital transformation is characterized by the operational and digital initiatives the corporation takes on, its technology adoption, and its corporate characteristics which include its culture, skillset, objectives, policies, and risk management. Figure 1 shows these dynamics and how they affect each other.

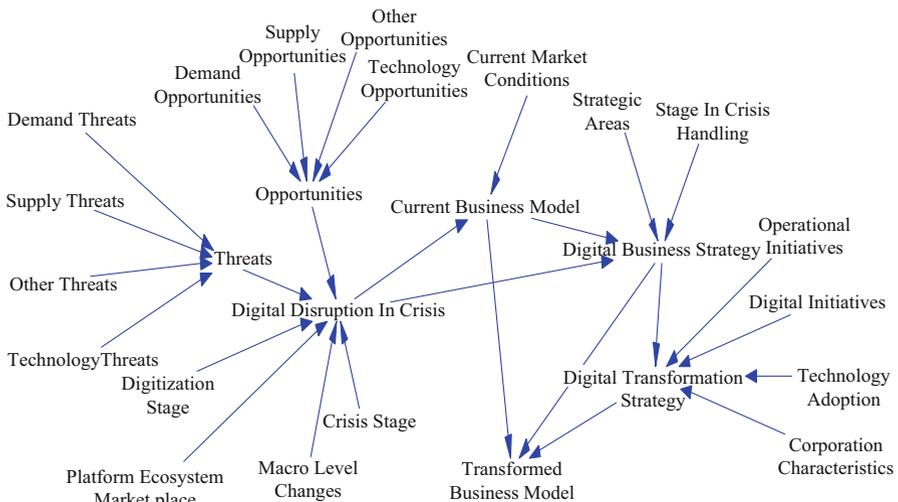


Fig. 1 Dynamics to consider for digital disruption in crisis

2 Framework to Model the Digital Business Strategy During Crisis

Given so many complex dynamics involved, we believe an intelligence framework should be used to model the digital business strategy. The intelligence framework should be (1) layered to provide decision-making help at multiple levels including business-level objectives, strategy- and policy-level objectives, and functional objectives; (2) should support qualitative analysis to model uncertainty situations where qualitative analysis with hypothesis is helpful; (3) should support sensitivity analysis; and (4) should support incorporating various strategy evaluation and characterization methodologies such as SWOT analysis, PEST (political, economic, social, and technological) analysis, Porter's five forces analysis, and Pareto analysis. We presented such an intelligence framework that employs system dynamics (SD) modeling together with economical, statistical, and machine learning models and outlined how to use it for various complex and uncertain situations in our earlier work [7–19].

Characterization of the involved dynamics is essential in the framework: the corporation should characterize the dynamics in a holistic way to understand the nature of the crisis and the disruption it causes. With a holistic view, executives can better understand the threats, see their opportunities more clearly, and devise more effective digital business and digital transformation strategies. The first step in the framework is to characterize the relevant dynamics. In the next sections, we will outline the dynamics involved in digital business strategy during crisis and how the corporations could characterize them.

3 Characterization of Crisis

First thing to do is understanding and characterizing the crisis which is important but not easy at all. The basic definitions and approaches we outlined about in our digital disruption work in [2–4] may apply here or may not necessarily apply during crisis. If our traditional economy and even digital economy assumptions and approaches do not necessarily apply, how could corporations to go through this crisis?

For the characterization of crisis, corporations should perform the following exercises.

1. Corporations should watch the macro level dynamics and try to understand the changing fundamentals. The crisis will bring in many vulnerabilities and threats but on the other hand could bring opportunities for the corporation. Corporations should characterize the changing fundamentals in demand, supply, market structure, and overall digital trends.
2. Corporations should watch the phase of the crisis to develop better aligned plans and should take transformational initiatives for a good result. And hopefully that

good result would be higher resiliency to future crisis, better productivity, and better delivery to customers.

Regarding the second item, which is about the phase of crisis, there are two aspects to this characterization. The first one is to characterize the stage of the crisis itself, and the second one is the stage in which the corporation is handling the crisis. We will address the first aspect below in this section and the second aspect in “Stage of Crisis Handling” section.

[1] identifies five distinct phases for crisis analysis: pre-crisis normality, emergence, occurrence, aftermath, and post-crisis normality. Each phase differs in its content, duration, and managerial opportunities, and the phases are not necessarily linear but circular. The corporation should perform an exercise to characterize the dynamics of these phases, formulate how to incorporate the phases into their decision-making, ensure their future preparedness, and learn how to prepare for and predict potential future crises [1].

Every corporation deals with unique challenges during the COVID-19 crisis, like some trying to survive while some others reaching their new highs. Therefore, there is no single outcome of these exercises. In the subsections, we will elaborate on further characterization of the crisis.

3.1 Characterization of Macro Level Changes

This is an exercise where the corporation needs to evaluate the macro level changes in economies and tries to characterize them. How the macro-economy will look like? Although there are many uncertainties, macro level changes during this crisis are showing mostly different consumer behaviors; new ways of working, education, health, conducting business, and all; changing industry structures; and value propositions being redistributed across existing and new ecosystems. Economies are facing possibility of going into recession. Small businesses are struggling more. All counties and everyone are affected, but in different ways. We will likely experience rising unemployment, shuttered businesses, corporate failures, credit defaults, falling asset prices, market volatility, and financial system vulnerabilities.

At the high level, we can characterize the macro level changes by the following aspects:

- *Changing industry structures and consumer behavior*: One is changing industry structures and consumer behavior including individual consumers and corporate customers. We can say the crisis accelerated the move to online, digital, and virtual for both individual and corporate customers.
- *Distance*: Another one is distance, which means we may see less human interaction, less travel especially international travel, more preference for local over global products and services, less cross-border trade, less global supply chain, more interest in producing and sourcing closer to end markets, and possibly more resistance globalization. These are all expected to impact the

corporations. We will most likely see an economy which favors less contact. Digitization already led to a smaller contact economy, for example, with digital commerce, telemedicine, and tele-everything. COVID-19 is accelerating this trend dramatically and with more automation such that businesses could produce and deliver products and services to the end customer with little or no human contact.

- *New ways*: Another difference is new ways of doing things. Consumers are doing things differently. Businesses are trying to find new ways to operate for resiliency and for efficiency; new forms of supply chains; and new plans for backup and safety, and more than finding new ways, they are reimagining their business models.
- *Government involvement*: Another one is more government involvement. Governments already took measures, and they are expected to exercise greater control on the economy by directing spending to help with citizens' health and well-beings, preserving jobs, and helping businesses keep alive. We may see more nationalization of products deemed essential for national security and governments taking equity stakes in businesses; providing stimulus, subsidies, tax breaks, and loans; and introducing more regulations in some industries.

3.2 Characterization of Disruption by Crisis

For any corporation, this is an exercise to characterize how the crisis will likely impact their business.

3.2.1 Demand-Side Characteristics

Characterizing the demand-side changes, threats, and opportunities is significant in overall characterization of disruption by the crisis. Since every individual is affected, some changes are inevitable in our behavior and activities, which will impact the demand side in economies. Certainly, our immediate need is safeguarding our lives and livelihoods, and that impacts our behavior as consumers. During such times, people question a lot of things: even life, purpose, meaning, priorities, and what is more important. Consumer behavior is hard to predict given so many uncertainties, but we can make some observations below, which are subject to critique:

- We saw initially a dramatic drop in economic activities. People stopped shopping except for essentials and stopped travel, restaurants, and cut-on purchases that can be postponed such as cars and appliances but increased shopping for items deemed essential and did more precautionary saving in the fear of unknown. Many people could not work and make earnings and therefore reduced their spending. We now see economies are trying to recover with opening up the businesses and going back to some normal.

- Online sales increased. Online activities in general increased with tele-work, tele-education, tele-medicine, tele-socialization, and all. Importance of telecom technologies, the Internet, and mobile and all relevant technologies has been even more realized during the crisis. We saw use of telecommunication services more than ever.
- Regarding consumer information needs, we saw an increase in seeking COVID-19-related content online and increase in consuming news.
- On consumer entertainment needs, we can observed moving to streaming and OTT even more, and even more of cutting cord particularly due to lack of sports. As for music, we saw personal concerts and expect digital performances of artists without audience becoming common. As there are no public sport events and live sport shows yet, consumers instead consume more online entertainment and online gaming.
- Consumers reduced expenses keeping more cash reevaluating their investment. With re-openings of the economies, we might see less participation in sharing economy for some products. People may want to own the items they see essential rather than renting or sharing. For example, people may prefer car ownership than relying on ridesharing or public transportation or depending on others providing those services, similarly for other things considered essential for being more self-sufficient and less dependent.
- Consumers may become more health savvy and change their behavior of consuming food, exercise, and healthcare services.
- Another possible change in consumer behavior could be to continue the practice of physical distance and privacy, which may influence where and how they want to live. People may question the value of the location in real estate if they consider working remotely will be the new normal. For example, rather than living in high expensive areas and paying huge rents, they could live away of their offices and still work remotely.
- On the other hand, some consumers are willing to give up on their digital privacy for the sake of public health. An example is the willingness to use COVID-19 tracking apps. Although such tracking apps ensure user anonymity, some consumers are willing to use them even if the apps didn't provide anonymity, while others do not consider using them due to privacy concerns. The debate between public health and privacy will continue.
- Consumers pay more attention to sustainability and social responsibility efforts of corporations during COVID-19 pandemic. They demand the corporations do good for the communities in which they operate.

Regarding corporate customers, we can list some changes on their demand:

- We see reduction in spending of corporations whose primary objective is to survive. Cash preservation is a common practice.
- We see more of the pay-as-you-go model and keep money rather than writing big checks, for example, in use of even more cloud services as opposed to spending for on-premise IT infrastructures.

- We expect more investment into improving the remote work as opposed to investing into corporate offices including corporate IT infrastructures, and we may see even questioning of paying high fees for commercial real estate for offices as the shift to work from home continues.
- Each corporation is different, but we expect speeding up the digital transformations and increased demand towards products, services, and digital platforms which help the corporations with their digital transformations.

Certainly things could change on the demand side as we progress in different phases of the crisis.

3.2.2 Supply-Side Characteristics

Similarly to demand side, corporations should perform an exercise to characterize the supply-side changes, threats, and opportunities. On the supply side, we can make some observations:

- We expect more production and sourcing moving closer to the end user.
- We expect more use of technology to keep the productivity when labor is unavailable or not adequate.
- Due to national interests, we expect manufacturing and service of essentials to be done more within national borders as opposed to being supplied by different countries.
- We expect turning more products into services and expect organizations using more of the online and virtual services to provide their own services, like universities offered their classes online.

3.2.3 Further Characterizations of the Crisis

Further characterization of crisis disruption includes the following aspects:

- Market structure-related changes
- New market-making-related changes
- New value proposition-related changes
- Reimagined business systems-related changes
- Platforms and ecosystems-related changes
- Disruptive technology-related changes
- Cross-boundary disruption changes
- Policy changes

The corporation should perform exercises to characterize the disruption due to crisis with respect to the above aspects similarly to characterizations outlined for digital disruption in our earlier work [2–4], and therefore we will not cover these characterizations in this paper. We will briefly mention few things:

- *Platforms and ecosystem-related changes*: Ecosystems will likely be challenged due to changes in the supply chain as we outlined in “Supply-Side Characteristics.”
- *Disruptive technology-related changes*: The corporation should carefully monitor for new low-end disruptive technologies, for example, what new technologies for health, consumer behavior, new ways of working, new policies, and new supply due to crisis could come and could disrupt. Disruptive technologies appear due to four main reasons: The first one is regulatory changes, for example, going from regulated to deregulated. The second one is supply-side factors such as changes in production technology leading to cheaper and faster production or changes in distribution and transaction costs which makes it easy to sell online. The third one is demand-side factors like changes in consumer behavior and affinity for new features which lead to creation of new markets. The last one is about change in information structure or coordination costs, for example, in the form of product/service reviews and ability to coordinate much more easily such as in crowdsourcing and sharing economy. The COVID-19 crisis makes all these conditions imminent for new disruptive technologies, particularly for low-end disruption.
- *Cross-boundary disruption changes*: We can say that the crisis will likely open some industries to further cross-boundary disruption as those industries face major shifts and are very vulnerable to cross-boundary disruption but probably less possibility of cross-boundary disruption coming from outside of national boundaries on some industries due to increased domestic productions and supply chains.
- *Policy changes*: We expect some policy changes on healthcare, education, business, and broadband access. Regarding economy, we mentioned that governments may be more involved in economy. We can expect policies to prevent widespread bankruptcies, support impacted citizens financially, and protect the financial system and the more affected sectors economically. Questions arise regarding policy changes, for example, what kinds of policies to develop and how society will be impacted. Another reason for policy changes could be on national security-related policies, like manufacturing of essential items nationally and to be more protective of the nations’ critical infrastructures during crisis.

4 Characterizations of Relevant Dynamics

The corporation should characterize other dynamics relevant to the disruption of the crisis including the ones below:

- Corporate objectives with the digital business strategy during crisis
- Characterizations of expected benefits and drawbacks of the new digital business model

- Corporate characteristics including culture, skills, leadership, and stage at addressing the crisis
- Corporate strategic areas to focus during crisis
- Corporate initiatives to deal with the crisis
- Alignment of corporate policies and strategies with the new digital business strategy
- Characterization of risks with the new digital business model and mitigation plans
- Characterization of corporate digital technology adoption during crisis
- Characterization of costs related to the new digital business strategy
- Characteristics of success metrics for the new digital business strategy

We addressed most of the above characterizations in the context of digital disruption in [2–4]. In this paper, we will further cover some of the above characterizations below in the context of crisis.

4.1 Corporate Strategic Areas

The corporation should do an exercise on choosing and prioritizing the strategic areas to tackle in order to come back stronger. Some strategic areas to focus on include recovering revenue, rebuilding operations, rethinking the organization, and accelerating the adoption of digital solutions [20]. Then, the corporation should develop actions based on where it is in handling the crisis on these strategic areas which we will talk about next. Characterization of corporate strategic areas involves an exercise of evaluating the importance of the focused areas and corporation's effectiveness on fulfilling the requirements of these areas.

4.2 Stage of Crisis Handling

This is an exercise where the corporation needs to determine which state it is in handling the crisis and evaluate its effectiveness in handling the crisis at that stage. The corporation may go through five stages as pointed out in [20]. The corporation should determine the actions suitable for the state it is in and for the strategic areas it is focusing on.

1. Resolve: An example action at this stage could be implementing business continuity and employee safety plans with support of remote work.
2. Resilience: An example action at this stage could be taking care of cash management in near time but meanwhile planning for longer-term economic and social sustainability.

3. Return: An example action at this stage could be to reactivate the supply chain, even with modifications to it and return the business to effective production at pace and at scale.
4. Reimagination: After experimenting the previous stages and by characterizing the many dynamics we outlined, the corporation should rethink about its vulnerabilities and opportunities to improve the performance of businesses, and reinvent itself if necessary, through a transformation.
5. Reform: After reimagining, the corporation should perform necessary transformations to the new itself. An example action at this stage is to define a corporate social responsibility to give back to the community in line with the corporation's purpose, its core reason for being, and its impact on the communities it operates in.

4.3 Characterization of Corporation

Corporation's culture is an important aspect to characterize and evaluate during crisis: The cultural challenges are big, and new skillsets are needed during the COVID-19 crisis. For example, working from home requires new ways of collaboration among employees, customers, and partners for development, sale, marketing, customer engagement, and support, similarly for skillset.

Leadership of the corporation is an important aspect to characterize. Leaders could excel in managing the business during the difficult times but also excel by providing safety to employees, providing employees timely and honest information about where the corporation is and where it is heading even if they don't have the complete picture, and in establishing trust.

The corporation should characterize its strengths and weaknesses in these areas.

4.4 Corporate Transformation Initiatives

Corporations should device initiatives based on its objectives and its characterization of dynamics. Corporations should engage in change and transformation proactively during crisis whether the change is about shaping markets, designing innovative solutions, or using middle managers as change agents [1]. The corporation should actively take a part in society during crisis, for example, with corporate social responsibility and philanthropic initiatives, as common threats could be better addressed together by thriving as a society. The corporation should consider collaborating actively on increasing the shared value with partners and event competitors [17–19].

Although transformation initiatives are many, we will look into two categories of transformation initiatives:

1. First is initiatives in operations, some for short-term crisis management and some for longer term to adjust to the new normal.
2. Second is the digital initiatives which may lead to revised overall digital strategy including the digital transformation strategy.

We will elaborate these two categories of initiatives below.

4.4.1 Operational Initiatives

The corporation should take some operational initiatives in line with the stage it is in with respect to handling the crisis. Following are some initiatives to consider during crisis [20]:

- Enabling business continuity which may be challenging
- Organizing to respond to crises by building a network of teams with some common goals and priorities set forth by the leaders
- Making decisions amid uncertainty and many unknowns
- Demonstrating empathy by dealing with human tragedies
- Communicating effectively by maintaining transparency and providing frequent updates

Characterization exercise for operational initiatives involves evaluating the impact of the initiatives and corporation's effectiveness in succeeding in them.

4.4.2 Digital Initiatives

In addition to taking initiatives for daily operations, the corporation should consider taking digital initiatives to be led by the executives. These digital initiatives could lead to defining the new digital business model of the corporation in the new normal. The C-level executives including CIO, CTO, CDO, CFO, CMO, and CEO should take part in defining the new digital business model and taking on digital initiatives.

Digital Initiatives by CIO

Corporate infrastructures are supporting the replacement of workers; in doing so, they are being challenged by various security attacks. CIO could take some digital initiatives during crisis to drive the transformation and to help the operations [20].

- One is to make remote working possible including collaboration tools, video-conferencing support, and training opportunities for employees. This initiative involves making sure the VPN servers are capable of many remote workers. In addition to technical initiatives, CIO could take cultural initiatives to make remote working effective.

- Related to first initiative is driving adoption of new ways of working among employees, partners, and customers.
- Another related initiative is to stabilize critical infrastructure, systems, and processes to support remote workers and new ways of working.
- Another initiative is being proactive on security and defending against security attacks that may try to take advantage of the crisis and exploit vulnerabilities due to new ways of working.
- Another initiative is to help with cash preservation by going more with pay-as-you-go models for services supporting new ways of working as opposed to big payments into infrastructures.
- A major initiative is to enable the shift in business processes for transformation which spans over intranets, data centers, websites, consumer-facing apps, and call centers.

Digital Initiatives by Other Executives

Corporations face many challenges in their digital transformation during the COVID-19 crisis. Some questions in addition to questions we asked before include [20]:

- How to understand customers and their businesses and needs, and then how to support them during crisis
- How to allow safe interactions among employees, partners, suppliers, and customers in new ways of working together, and how to digitize interactions among them
- How to help transition customers to online channels
- How to maintain the customer relationships
- How to empower development, sale, marketing, customer engagement, and support teams, enhance their autonomy, and promote decentralized decision power
- How innovative solutions can be rapidly delivered as minimum viable products (MVPs) to meet immediate customer demands
- How to ensure data protection and privacy and information security especially when the corporation is exploring new ways of working that didn't experience before
- What infrastructure is needed to support the development of new digital channels and customer experiences
- How to take advantages of technologies to achieve digital initiatives
- What should be the new digital business model, how to define it, and how to transition to it

All these C-level executives CTO, CDO, CFO, CMO, and CEO could work together to identify digital initiatives in order to reach the new digital business model. Some initiatives to consider include [20]:

- Rebalancing the product/service road map in line with the new digital business strategy
- Engaging with customers in new ways
- Updating agile practices to accelerate remote delivery
- Building up cash reserves
- Accelerating digital ambition, analytics engines, and automation by deploying more AI and software-defined components
- Becoming ready to capture early demand

Characterization exercise for digital initiatives involves evaluating the impact of the initiatives and corporation's effectiveness in succeeding in them.

5 Developing Models for Digital Business Strategy During Crisis

In our previous work, we developed a generic model for modeling digital business strategy and evaluating digital transformation strategies in [2–4] and discussed the challenges with modeling it. Modeling the digital business strategy during crisis is even more challenging due to more uncertainties, stronger and sudden threats, and many more dynamics to consider. Figure 1 shows the dynamics that should be considered for modeling this strategy. The framework in [2–4] suggests developing systems dynamics (SD) models after the characterization phase. Figure 1 should serve as a starting point for building an SD model. The characterization exercises should yield some formal parameters and representations since they will be analyzed using tools. The parameters from characterization exercises should be inputted into the model as arrays of parameters for each variable where each variable corresponds to a different aspect of characterization. The SD model should have stock variables to measure the aggregated benefits of the digital business strategy to the corporation. Causal loops should be introduced to incorporate the hypothesis and to simulate the network effects. SD model is only one part of modeling. Economical, statistical, and machine learning models should augment the SD model. Sensitivity analysis should be done to model various strategic options and decisions under uncertainties, for which SD modeling offers good advantages. This paper builds the foundation and defines requirements for building a generic SD model for digital business strategy during crisis by characterizing the relevant dynamics. Developing a generic SD model is our further step in this research.

6 Conclusion and Future Work

COVID-19 is a major health, economic, and social crisis in the modern age. How exactly it will evolve remains to be seen as there is still big uncertainty. Even before the COVID-19 pandemic, digitization had changed the consumer

behavior and habits, regulations, supply-side factors, demand-side factors, and costs of information structure and coordination. COVID-19 will likely accelerate the digitization but will also force corporations to refine and possibly redefine their digital business strategies.

Corporations have to address many questions about how to deal with the crisis in digital age and refine their digital business strategy while going through some transformations already. The main question is how corporations could navigate through this crisis when traditional economy and even digital economy assumptions and approaches do not necessarily apply. Corporations need to understand the relevant dynamics and characterize them in a holistic view. Macro level changes, stage of crisis, digital disruption during crisis, digital marketplaces, current market conditions, current business model of the corporation, digital transformation strategy, and transformed business model of the corporation should be characterized and should be considered in building the models for evaluating the digital business strategy during crisis. Due to many uncertainties, complex dynamics involved, and complex relationships among them, modeling the digital business strategy during crisis is very challenging. We believe a layered intelligence framework that facilitates abstracting the macro level dynamics and corporate objectives at multiple levels, including business-level objectives, strategy- and policy-level objectives, and functional objectives, is helpful for decision-making. When all the dynamics and factors are integrated in a layered intelligence framework, executives can use the resulting models to better understand how different dynamics could impact their business, supply chain, industry, market, digital transformation, and ecosystem as well as how different strategic options would play out in the transformed business model to overcome the crisis.

In this paper, we study how corporations could characterize the digital business strategy during crisis and devise a framework for how they could model and evaluate various strategic options. Due to so many complex dynamics involved and many uncertainties, we argue a layered intelligence framework that supports qualitative analysis that should be used to model the digital business strategy during crisis. This paper builds a foundation for our research on digital business models during crisis by characterizing the dynamics. Our future work includes building models for the digital business strategy and deriving insight from them.

References

1. C.L. Pedersen, T. Ritter, C.A. Di Benedetto, Managing through a crisis: Managerial implications for business-to-business firms. *Ind. Market. Manag.* **88**, 314–322 (2020). <https://doi.org/10.1016/j.indmarman.2020.05.034>. Epub 2020 Jun 5. PMID: PMC7273163
2. S. Yucel, Modeling digital transformation strategy, in *5th Annual Conference on Computational Science & Comp Intelligence (CSCI'18)*, 13–15 Dec 2018, Las Vegas, USA
3. S. Yucel, Estimating the benefits, drawbacks and risk of digital transformation strategy, in *5th Annual Conference on Computational Science & Comp Intelligence (CSCI'18)*, 13–15 Dec 2018, Las Vegas, USA
4. S. Yucel, Modeling digital business strategy, in *5th Annual Conf. on Computational Science & Computational Intelligence (CSCI'18)*, 13–15 Dec 2018, Las Vegas, NV, USA
5. McKinsey and Company, The economic essentials of digital strategy. <http://www.mckinsey.com/business-functions/strategy-and-corporate-finance/our-insights/the-economic-essentials-of-digital-strategy>
6. D. Goerzig, T. Bauernhansl, Enterprise architectures for the digital transformation in small and medium-sized enterprises, Elsevier. *Procedia CIRP* **67**, 540–545 (2018)
7. S. Yucel, Delivery of digital services with network effects over hybrid cloud, in *The 12th International Conference on Grid, Cloud, and Cluster Computing, GCC'16*, 25–28 July 2016, Las Vegas, NV, USA
8. S. Yucel, Evaluating different alternatives for delivery of digital services, in *The 12th International Conference on Grid, Cloud, and Cluster Computing, GCC'16*, 25–28 July 2016, Las Vegas, NV, USA
9. S. Yucel, I. Yucel, A model for commodity hedging strategies, in *The 13th Int. Conference on Modeling, Simulation and Visualization Methods (MSV'16)*, 25–28 July 2016, Las Vegas, USA
10. S. Yucel, I. Yucel, Estimating the cost of digital service delivery over clouds, in *The 2016 International Symposium on Parallel and Distributed Computing and Computational Science (CSCI-ISPDP)*, 15–17 Dec 2016, Las Vegas, NV, USA
11. S. Yucel, Smart community wireless platforms, in *The 14th International Conference on Modeling, Simulation and Visualization Methods (MSV'17)*, 17–20 July 2017, Las Vegas, NV, USA
12. S. Yucel, Measuring benefits, drawbacks and risks of smart community wireless platforms, in *The 14th Int Conference on Modeling, Simulation and Visualization Methods (MSV'17)*, 17–20 July 2017, Las Vegas, USA
13. S. Yucel, Estimating cost of smart community wireless platforms, in *The 14th Int Conference on Modeling, Simulation and Visualization Methods (MSV'17)*, 17–20 July 2017 Las Vegas, USA
14. S. Yucel, Smart city wireless platforms for smart cities, in *The 14th International Conference on Modeling, Simulation and Visualization Methods (MSV'17)*, 17–20 July 2017 Las Vegas, USA
15. S. Yucel, Evaluating enterprise mobility strategy, in *4th Annual Conference on Computational Science & Computational Intelligence (CSCI'17)* | 14–16 Dec 2017, Las Vegas, NV, USA
16. S. Yucel, Estimating cost of enterprise mobility strategy, in *4th annual Conference on Computational Science & Computational Intelligence (CSCI'17)*, 14–16 Dec 2017, Las Vegas, NV, USA
17. S. Yucel, Modeling corporate social responsibility strategy, in *CSCE'18 - The 2018 World Congress in Computer Science, Computer Engineering & Applied Computing*, 30 July–02 Aug 2018, Las Vegas, USA
18. S. Yucel, Estimating cost of CSR strategy, in *CSCE'18 - The 2018 World Congress in Computer Science, Computer Engineering, & Applied Computing*, 30 July–02 Aug 2018, Las Vegas, NV, USA

19. S. Yucel, Measuring benefits, drawbacks and risks of CSR strategy, in *CSCE'18 – The 2018 World Congress in Computer Science, Computer Engineering, & Applied Computing*, 30 July–02 Aug 2018, Las Vegas, NV, USA
20. McKinsey and Company, The path to the next normal. <https://www.mckinsey.com/~media/McKinsey/Featured%20Insights/Navigating%20the%20coronavirus%20crisis%20collected%20works/Path-to-the-next-normal-collection.pdf>

Dealing Bridge Hands: A Study in Random Data Generation



Peter M. Maurer

1 Introduction

Recently, we have embarked on a massive upgrade of the DGL (data generation language) software [1, 2], both to support modern types of software development and to enable the software to handle problems that were difficult or impossible using the original language structures. In some cases, we simply added features that we felt were missing, and in some cases we chose a specific problem and added the features necessary to solve the problem. The objective in both cases was to provide a more effective data generation tool without damaging the existing effectiveness of the language. The problem of dealing bridge hands was used to add several new features to DGL. Because DGL language structures are productions, which themselves are completely modular, adding these new features could be done without changing any existing features of the language. In this paper, we go through the development process that led to several new language features. These new features can be extremely useful for more substantial problems as well.

2 Basic Bridge Hands

The original DGL features allow one to generate bridge hands with the cards in random order. The first step is to create the model of a deck of cards. One such method is to simply list the name of each card in an ordinary production as in Fig. 1.

P. M. Maurer (✉)

Department of Computer Science, Baylor University, Waco, TX, USA

e-mail: peter_maurer@baylor.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_70

961

```
Deck:      "Ace of Spades", "King of Spades", "Queen of Spades", ...
           "4 of clubs", "three of clubs", "two of clubs" ;
```

Fig. 1 A card deck model

```
Fig. 2 An improved card      Number: macro Ace,King,Queen,Jack,10,9,8,7,6,5,4,3,2;
deck model                    Suit: macro Spades,Hearts,Diamonds,Clubs;
                               Deck: !Number " of " !Suit;
```

```
Fig. 3 The deck model      Number: macro Ace,King,Queen,Jack,10,9,8,7,6,5,4,3,2;
                               Suit: macro Spades,Hearts,Diamonds,Clubs;
                               Deck: unique restart !Number " of " !Suit eol;
```

```
main:      North eol %13{Deck} eol East eol %13{Deck}
           eol West eol %13{Deck} eol South eol %13{Deck} "-----" eol ;
Number: macro Ace,King,Queen,Jack,10,9,8,7,6,5,4,3,2;
Suit: macro Spades,Hearts,Diamonds,Clubs;
Deck: unique restart !Number " of " !Suit eol;
```

Fig. 4 Bridge hands version 1

A more effective method is to use macros as shown in Fig. 2.

An ordinary production will permit each card to be selected several times. This is unacceptable, so the keywords *unique* and *restart* must be added to the deck production as in Fig. 3. The restart keyword is necessary if multiple sets of hands are to be generated. We also want each card to appear on a separate line in the output, so we add an end-of-line character to each card name. We had observed that end-of-line characters were necessary for many data generation tasks, so we added a new key word, *eol*, to generate this character more easily.

To generate the complete set of hands, it is now necessary to create a main production giving the format of a set of hands. This production is shown in Fig. 4 along with the deck model of Fig. 3.

3 Simple Sorting

This method gives bridge hands that are correct but are in random order. This can be a problem if someone wishes to use the output for a duplicate bridge tournament where several tables must be dealt identical hands. This is not really practical unless the cards in each hand are sorted into suit and rank order.

To solve this problem, we created the *sorter* production. The *sorter* production is based on the existing *queue* production, but unlike the *queue*, the *sorter* may not have initial values. Strings can be added to a *sorter* production using a non-terminal of the form *%{a.b}* where b is the name of the *sorter* production and a is either the

string to be added or the name of a production that generates the string. A *sorter* production can contain an arbitrarily large number of strings. A production of the form %b or %{b} removes the smallest item in the *sorter* production and generates it.

Strings are sorted using normal string-sorting conventions. There are two possible approaches to sorting which are sort-on-input and sort-on-output. For sort-on-input, the internal list of strings is maintained in sorted order. When a new string is added, it is inserted into the list in the correct position. This is the insertion sort algorithm which is known to be worst case $O(n^2)$. For sort-on-output, the list is maintained in random order, corresponding to the order in which the items were originally added. Upon the first output (%b or %{b} non-terminal), the list is sorted using any convenient algorithm. This permits efficient algorithms such as quicksort to be used. A “dirty” flag must be kept to indicate whether the list has been sorted since the last string was inserted. This flag prevents unnecessary sorts and permits string insertions to be intermixed with string generation. We ultimately chose sort-on-input because it is simpler to implement, but this may change in the future.

Of course, simple sorting of the card names will not yield hands in the correct order. To accomplish this, it is necessary to change the declaration of the deck so that each card name is preceded by its position in the deck. Figure 5 shows how this is done.

A sorter production must be added, and the main production must be modified as shown in Fig. 6.

Simple sorting gives the cards in correct order, but the final output has a sorting number in front of each card. It would be nice if this number could be eliminated, since it gives no information about the card. There are several ways of doing this.

```
Deck: unique restart
" 1. Ace of Spades\n", "13. 2 of Spades\n", "12. 3 of Spades\n", "11. 4 of Spades\n",
"10. 5 of Spades\n", " 9. 6 of Spades\n", " 8. 7 of Spades\n", " 7. 8 of Spades\n",
...
"14. Ace of Hearts\n",
...
```

Fig. 5 A deck for simple sorting

```
Sort: sorter;
main: North eol %13{Deck.Sort}%13{Sort} eol
      East eol %13{Deck.Sort}%13{Sort} eol
      West eol %13{Deck.Sort}%13{Sort} eol
      South eol %13{Deck.Sort}%13{Sort} "-----" eol ;
```

Fig. 6 Simple sorting main production

4 Keyed Sorting

Our first approach to eliminating the sorting numbers from the output was to create a production, *ksorter*, that maintains a separate of sort key for each item. The items are sorted using their sort keys, but when generating items, only the item itself appears. The keys remain invisible. The main problem with this is that sort keys must be assigned at the same time as the item, and in the original DGL, there is no way to do this.

To solve this problem, we created the *keyed* production, which contains pairs of items, a key followed by a data item. If there are an odd number of items, the last item is considered a data item with the null string as its key. The *keyed* production permits items to be selected multiple times, but we also provided a *unique keyed* production to permit selection without replacement. If either *keyed* production is used as an ordinary production, it will be as if the keys did not exist. The only time the behavior is different from an ordinary production is when a *keyed* production is assigned to a *ksorter* production. When a non-terminal of the form $\%{a.b}$ is interpreted, where *a* is a *keyed* production and *b* is a *ksorter* production, both an item and its associated key are selected from *a* and assigned to *b*. Figure 7 shows how *keyed* productions and *ksorter* productions can be used to deal bridge hands.

5 Indexed Sorting

Keyed sorting gives a complete solution to the problem, but the solution is somewhat inelegant. The *keyed* and *ksorter* productions are rather clumsy, although they may be useful in some applications. Further development has provided several other approaches to the problem, some of which are much cleaner than *keyed* sorting. One new development was the addition of commands to non-terminals of the form $\%{a.b}$. This new model permits *a* to be a command rather than a string or the name of a non-terminal. Initially commands were parameterless,

```
main:      North eol %13{Deck.Sort}%13{Sort} eol
          East eol %13{Deck.Sort}%13{Sort} eol
          West eol %13{Deck.Sort}%13{Sort} eol
          South eol %13{Deck.Sort}%13{Sort} "-----" eol ;
Sort: ksorter;
Deck: unique keyed restart
" 1","Ace of Spades\n", "13","2 of Spades\n", "12","3 of Spades\n", "11","4 of Spades\n",
"10","5 of Spades\n", " 9","6 of Spades\n", " 8","7 of Spades\n", " 7","8 of Spades\n",
" 6","9 of Spades\n", " 5","10 of Spades\n", " 4","Jack of Spades\n",
" 3","Queen of Spades\n", " 2","King of Spades\n", "14","Ace of Hearts\n", "26","2 of Hearts\n",
...
```

Fig. 7 Keyed sorting

such as in the command `%{$count.b}` which generates the number of items in *b*. (All commands start with the character `$`.) Although parameterless commands significantly increased the power of DGL, an even more significant improvement was the addition of command parameters, such as `%{$length(3).b}` which generates the length of alternative 3 of production *b*. With commands, it is possible to assign keys to the data items of a *ksorter* production without using *keyed* productions. The command `%{$key(3,k1).b}` assigns the string “k1” as the key of item 3 in the *ksorter* production *b*.

Another new idea was indexed selection for normal productions which was originally developed to permit synchronization between different normal productions. Indexed selection uses non-terminals of the form `%{n.b}`, where *b* is a normal production *b* and *n* is a number that can be explicitly specified or generated by a production. Such a non-terminal selects the *n*th alternative from production *b*, where the first alternative is alternative 0.

Yet another new idea was the idea of integer sorters (*isorter*) and floating-point sorters (*fsorter*). The *sorter* production sorts using a string sort. Sometimes it's necessary to sort integers or floating-point numbers, neither of which can be easily sorted as strings. Combining integer sorting with indexed selection gives us a much simpler way of dealing bridge hands, which is shown in Fig. 8.

In Fig. 8, the production *Ix* has 52 alternatives, which are the numbers from 0 through 51. These alternatives are selected randomly without replacement. When a hand is dealt, 13 selections are made from *Ix* and assigned to the integer sorter *Sort*. These 13 values are sorted and used to index the ordinary production, *Deck*.

6 Trimming the Keys

After all the development of the two previous sections, a more straightforward solution was developed as part of a set of string manipulation features being

```

main: North eol %13 {Ix.Sort}%13 {Sort.Deck} eol
      East eol %13 {Ix.Sort}%13 {Sort.Deck} eol
      West eol %13 {Ix.Sort}%13 {Sort.Deck} eol
      South eol %13 {Ix.Sort}%13 {Sort.Deck}
      "-----" eol ;
Number: macro Ace,King,Queen,Jack,10,9,8,7,6,5,4,3,2;
Suit: macro Spades,Hearts,Diamonds,Clubs;
Deck: !Suit " " !Number eol;

Sort: isorter;
Ix: unique [0-9],[1-4][0-9],50,51;
    
```

Fig. 8 Indexed sorting

developed for the DGL package. This is the substring production, which can be used with simple sorting to give the desired result. Figure 9 shows how this is done. Figure 9 is identical to the simple sorting example, except *Trim* is referenced instead of *Sort* in the main production.

The idea of the substring production is that it represents a substring of a different production. The name of the other production follows the substring keyword, with one or two numbers following. The first number is the start position of the substring, and the second is the length of the substring. If the second number is omitted, the substring continues to the end of the string. Unlike most other DGL productions, the substring production is best viewed as an alternative interface to an existing production.

7 Experimental Data

When sorting bridge hands, a couple of different sort orders could be used. One such order is to rank the cards within a suit according to value and the suits according to strength. This would give the sort order in Fig. 10.

Bridge rank order would be typical of the way a player would arrange her hand (if she does so). However, if one is making up a number of identical hands out of new decks, it may be more advantageous to sort cards into new deck order. New

```
Sort: sorter;
main: North eol %13{Deck.Sort}%13{Trim} eol
      East eol %13{Deck.Sort}%13{Trim} eol
      West eol %13{Deck.Sort}%13{Trim} eol
      South eol %13{Deck.Sort}%13{Trim} "-----" eol ;
Trim: substring Sort,4;
Deck: unique restart ...
"41. King of Clubs\n";
```

Fig. 9 Trimming the sort key

Fig. 10 Bridge rank order

```
A,K,Q,J,10,9,8,7,6,5,4,3,2 Spades
A,K,Q,J,10,9,8,7,6,5,4,3,2 Hearts
A,K,Q,J,10,9,8,7,6,5,4,3,2 Diamonds
A,K,Q,J,10,9,8,7,6,5,4,3,2 Clubs
```

Fig. 11 Bicycle new deck order

A,2,3,4,5,6,7,8,9,10,J,Q,K Spades
 A,2,3,4,5,6,7,8,9,10,J,Q,K Diamonds
 K,Q,J,10,9,8,7,6,5,4,3,2,A Clubs
 K,Q,J,10,9,8,7,6,5,4,3,2,A Hearts

deck order can be different depending on the manufacturer. Some manufacturers arrange all suits in order from ace to king. Others do things differently as shown in Fig. 11 which gives new deck order for bicycle poker decks.

For simple sorting, keyed sorting, and trimmed simple sorting, deck order depends on the sort keys. For indexed sorting, order is identical to the order in which the card names are specified. In any case, care must be taken to insure cards appear in the desired order.

We have generated bridge hands using all methods and show the results below. Each method could be used to generate thousands or millions of different hands (Figs. 12, 13, 14, and 15).

The grammar given above needed to be altered slightly. The configuration of the macros shown above causes the aces to sort first, then the kings, and so forth. The macro references were reversed to cause spades to sort first, then hearts, and so forth (Fig. 16).

8 Conclusion

Although dealing bridge hands is a relatively lightweight application for DGL, the process of solving the problem in an elegant way has led to the development of a number of new useful features. This is one method of developing new useful features for DGL. Another effective method is to create new features first and then develop applications for them. Even when the second method is used, it is still useful to imagine how new features will be used when developing them. The latest version of DGL has dozens of new features that are not present in the original version. The new version of DGL should prove to be an effective tool for debugging modern software. In addition, DGL has been enhanced to the point where it can be considered to be a universal data generation tool. Research is under way to add new features to make sure that this remains true.

	North	
	5 of Diamonds	
	8 of Clubs	
	3 of Diamonds	
	6 of Diamonds	
	6 of Hearts	
	9 of Hearts	
	10 of Spades	
	6 of Clubs	
	Jack of Clubs	
	Queen of Clubs	
	10 of Hearts	
	4 of Clubs	
	Queen of Spades	
West		East
King of Clubs		5 of Hearts
5 of Clubs		6 of Spades
4 of Diamonds		10 of Clubs
Queen of Hearts		4 of Hearts
7 of Hearts		Ace of Hearts
3 of Hearts		Jack of Hearts
8 of Hearts		8 of Spades
10 of Diamonds		Ace of Diamonds
3 of Clubs		2 of Diamonds
King of Spades		Ace of Clubs
9 of Clubs		Jack of Diamonds
3 of Spades		2 of Clubs
King of Hearts		2 of Hearts
	South	
	9 of Spades	
	7 of Spades	
	Queen of Diamonds	
	4 of Spades	
	Ace of Spades	
	7 of Diamonds	
	King of Diamonds	
	5 of Spades	
	9 of Diamonds	
	8 of Diamonds	
	2 of Spades	
	Jack of Spades	
	7 of Clubs	

Fig. 12 Hands in random order

	North	
	1. Ace of Spades	
	2. King of Spades	
	3. Queen of Spades	
	5. 10 of Spades	
	8. 7 of Spades	
	11. 4 of Spades	
	17. Jack of Hearts	
	20. 8 of Hearts	
	24. 4 of Hearts	
	26. 2 of Hearts	
	27. Ace of Diamonds	
	35. 6 of Diamonds	
	50. 4 of Clubs	
West		East
9. 6 of Spades		7. 8 of Spades
10. 5 of Spades		12. 3 of Spades
13. 2 of Spades		14. Ace of Hearts
15. King of Hearts		19. 9 of Hearts
16. Queen of Hearts		29. Queen of Diamonds
21. 7 of Hearts		36. 5 of Diamonds
25. 3 of Hearts		39. 2 of Diamonds
32. 9 of Diamonds		41. King of Clubs
34. 7 of Diamonds		44. 10 of Clubs
38. 3 of Diamonds		45. 9 of Clubs
43. Jack of Clubs		47. 7 of Clubs
48. 6 of Clubs		49. 5 of Clubs
51. 3 of Clubs		52. 2 of Clubs
	South	
	4. Jack of Spades	
	6. 9 of Spades	
	18. 10 of Hearts	
	22. 6 of Hearts	
	28. King of Diamonds	
	2e. 5 of Hearts	
	30. Jack of Diamonds	
	31. 10 of Diamonds	
	33. 8 of Diamonds	
	37. 4 of Diamonds	
	40. Ace of Clubs	
	42. Queen of Clubs	
	46. 8 of Clubs	

Fig. 13 Simple sorted hands

	North	
	King of Spades	
	Queen of Spades	
	Jack of Spades	
	10 of Spades	
	6 of Spades	
	Ace of Hearts	
	Jack of Hearts	
	8 of Hearts	
	7 of Hearts	
	7 of Diamonds	
	6 of Diamonds	
	King of Clubs	
	2 of Clubs	
West		East
7 of Spades		8 of Spades
2 of Spades		5 of Spades
King of Hearts		4 of Spades
10 of Hearts		Queen of Hearts
9 of Hearts		5 of Hearts
6 of Hearts		3 of Hearts
4 of Hearts		Jack of Diamonds
Ace of Diamonds		8 of Diamonds
10 of Diamonds		4 of Diamonds
2 of Diamonds		3 of Diamonds
Jack of Clubs		Queen of Clubs
6 of Clubs		8 of Clubs
4 of Clubs		3 of Clubs
	South	
	Ace of Spades	
	9 of Spades	
	3 of Spades	
	2 of Hearts	
	King of Diamonds	
	Queen of Diamonds	
	9 of Diamonds	
	5 of Diamonds	
	Ace of Clubs	
	10 of Clubs	
	9 of Clubs	
	7 of Clubs	
	5 of Clubs	

Fig. 14 Keyed sorting

	North	
	Spades King	
	Spades 7	
	Spades 5	
	Spades 3	
	Spades 2	
	Hearts 10	
	Hearts 5	
	Hearts 2	
	Diamonds 10	
	Diamonds 6	
	Diamonds 3	
	Clubs 4	
	Clubs 3	
West		East
Spades Queen		Spades Ace
Spades Jack		Spades 10
Spades 9		Spades 8
Spades 6		Hearts Jack
Hearts King		Hearts 8
Hearts 9		Diamonds King
Hearts 4		Diamonds 8
Diamonds 9		Diamonds 7
Diamonds 5		Diamonds 4
Clubs Jack		Diamonds 2
Clubs 9		Clubs Ace
Clubs 7		Clubs Queen
Clubs 2		Clubs 6
	South	
	Spades 4	
	Hearts Ace	
	Hearts Queen	
	Hearts 7	
	Hearts 6	
	Hearts 3	
	Diamonds Ace	
	Diamonds Queen	
	Diamonds Jack	
	Clubs King	
	Clubs 10	
	Clubs 8	
	Clubs 5	

Fig. 15 Indexed sorting

	North	
	7 of Spades	
	6 of Spades	
	9 of Hearts	
	8 of Hearts	
	7 of Hearts	
	4 of Hearts	
	3 of Hearts	
	King of Diamonds	
	Jack of Diamonds	
	7 of Diamonds	
	3 of Diamonds	
	9 of Clubs	
	7 of Clubs	
West		East
10 of Spades		Queen of Spades
3 of Spades		Jack of Spades
2 of Spades		9 of Spades
King of Hearts		5 of Spades
6 of Hearts		4 of Spades
8 of Diamonds		Ace of Hearts
6 of Diamonds		Ace of Diamonds
Queen of Clubs		9 of Diamonds
Jack of Clubs		2 of Diamonds
10 of Clubs		Ace of Clubs
8 of Clubs		King of Clubs
6 of Clubs		3 of Clubs
4 of Clubs		2 of Clubs
	South	
	Ace of Spades	
	King of Spades	
	8 of Spades	
	Queen of Hearts	
	Jack of Hearts	
	10 of Hearts	
	2 of Hearts	
	Queen of Diamonds	
	5 of Hearts	
	10 of Diamonds	
	5 of Diamonds	
	4 of Diamonds	
	5 of Clubs	

Fig. 16 Simple sorting with trimming

References

1. P.M. Maurer, Generating test data with enhanced context free grammars. *IEEE Softw.* **7**(4), 50–56 (1990)
2. P.M. Maurer, The design and implementation of a grammar-based data generator. *Softw. Pract. Exp.* **22**(3), 223–244 (1992)

An Empirical Study of the Effect of Reducing Matching Frequency in High-Level Architecture Data Distribution Management



Mikel D. Petty

1 Introduction and Motivation

The High-Level Architecture (HLA) is an interoperability protocol standard and implementation architecture for distributed simulation. Using HLA, multiple concurrently executing simulation models collaborate to simulate a common scenario by exchanging data messages over a connecting network. In HLA, an individual simulation model is known as a “federate,” and a set of federates interacting in a simulation is known as a “federation.” In federations with many federates, very large volumes of messages are possible, potentially exceeding the individual federates’ host computers’ capacity to process the incoming messages. HLA’s Data Distribution Management (DDM) services provide a mechanism to reduce message volume and improve federation scalability. With DDM, federates declare both the possible types and values of data they will send and the types and values of data they wish to receive abstractly as axis-parallel hyper-rectangles in a multidimensional coordinate space. The HLA supporting software, known as the Run-Time Infrastructure (RTI), routes messages from a sending federate to a receiving federate if and only if their DDM rectangles intersect.

A common scheme for using DDM in entity-level combat simulation is to define a coordinate space corresponding to the geographical area of the scenario and define rectangles that correspond to the simulated entities’ locations (for sending data) and the area covered by their sensors (for receiving data). This implies that as the entities move in the simulation, their DDM rectangles will move in the coordinate space. Thus, to maintain the message routing connections, the RTI must repeatedly solve a computational geometry problem: given a set of axis-parallel hyper-rectangles in a

M. D. Petty (✉)

University of Alabama in Huntsville, Huntsville, AL, USA

e-mail: pettym@uah.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_71

975

multidimensional coordinate space, find all intersecting pairs of rectangles. In HLA DDM, this process is known as “matching.”

DDM matching can be computationally expensive, especially in large simulations with many moving entities. Time spent repeatedly performing matching is not available for other RTI functions. Consequently, a large amount of research effort has been devoted to developing data structures and algorithms to perform DDM matching as *efficiently* as possible. In this study, a different approach is taken; instead of (or perhaps in addition to) performing DDM matching as efficiently as possible, the approach studied here is to perform DDM matching as *seldom* as possible, as long as the results of the federation’s simulation are not compromised. To investigate this approach, an abstract constructive entity-level combat model, similar to production military semi-automated forces systems, was developed and verified. It was then used to experimentally assess the effect of performing DDM matching at various frequencies. Hostile battlefield entities can fire at each other only after sighting each other, and a sighting requires an exchange of messages between the federates simulating the entities, and an exchange of messages requires that the entities’ DDM rectangles intersect. Therefore, the effect of reducing the frequency of DDM matching was measured by tracking three times for each pair of hostile battlefield entities: when their DDM rectangles first intersected, when they first sighted each other, and when they first shot at each other.

This chapter is organized as follows. After this introduction, the next section provides explanatory background information on distributed simulation, the High-Level Architecture, and semi-automated forces systems. The following section details the research question addressed and procedure used to investigate it. The final section describes the test scenario, reports the results, presents the conclusions drawn from those results, and suggests possible future work.

2 Background

This section provides background information on distributed simulation, the High-Level Architecture, and semi-automated forces systems, which collectively are the context for the study reported in this chapter.

2.1 *Distributed Simulation, HLA, and DDM*

In distributed simulation, simulation systems are assembled from a set of independently executing models running on multiple computational nodes connected by a network. During execution, the models report the attributes (e.g., current location) and actions (e.g., firing a weapon) of interest regarding the entities they are simulating by exchanging network messages. An interoperability protocol defines the formats of the messages, the conditions under which messages should be sent,

and the proper processing for a received message. The High-Level Architecture (HLA) is both an interoperability protocol and an implementation architecture for constructing distributed simulation systems [2, 4, 19]. In HLA terminology, a set of collaborating models is a federation and each of the collaborating models is a federate. The HLA Run-Time Infrastructure (RTI) software provides services needed to support a simulation execution, including services to start and stop a simulation execution, to send simulation data between federates, and to coordinate the passage of simulated time among the federates [11].

One category of HLA services, Data Distribution Management (DDM), provides a publish-and-subscribe mechanism that can reduce the amount of data delivered to a federate (or equivalently, routed to the computer on which the federate is executing) during a simulation execution [14]. In DDM, a coordinate space with dimensions corresponding to user-selected simulation variables is defined. During execution, the federates declare what data values they expect to send and/or wish to receive by creating rectangular regions, called update and subscription regions, within the coordinate space. Update regions correspond to the limits of the variable values the publishing federate will send, and subscription regions correspond to the limits of the variable values the subscribing federate wishes to receive. Geometrically, a region is a single axis-parallel rectangular or hyper-rectangular subspace within the coordinate space. If an update region and a subscription region intersect in the coordinate space, then the data values to be sent by the federate associated with the update region and the data values to be received by the federate associated with the subscription region overlap, and therefore the RTI will deliver network messages from the publishing federate to the subscribing federate.

Figure 1 illustrates the basic ideas of DDM. In the lower half of the figure, a notional federation has three federates *A*, *B*, and *C*, represented by the ovals, connected by a network, and represented by the horizontal line. The upper half of the figure is the federation's DDM coordinate space; it has two dimensions, d_1 and d_2 . Each federate has declared one region in the DDM coordinate space. The update region declared by federate *A* represents the data that federate *A* will send. The subscription regions declared by federates *B* and *C* represent the data that those federates wish to receive. Federate *A*'s update region and federate *B*'s subscription region intersect, so updates to the data items associated with the update region are delivered by the HLA RTI from federate *A* to federate *B*. No data are delivered to federate *C*.

DDM coordinate spaces may have any number of dimensions. However, in the "most common" way of using DDM [7], a two-dimensional coordinate space is defined to correspond to the x and y coordinates of the geographical area in which the scenario is occurring (e.g., a battlefield). For each entity, the federate simulating it declares two regions: an update region centered on the entity's current location and sized so that the entity will not move out of it in the current time step, and a subscription region with its location and size determined by the entity's location and the area covered by the entity's sensors in the x and y dimensions. The intent of this scheme is that the federates will receive data regarding only those entities that their simulated entities could possibly detect with their sensors.

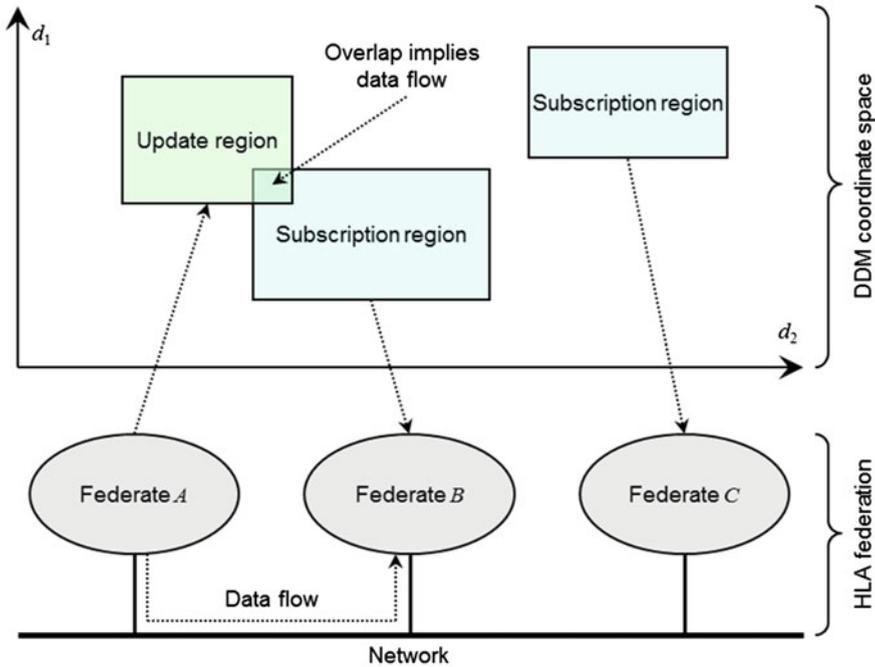


Fig. 1 A notional HLA federation and its DDM coordinate space

This use of DDM has an important implication. As the simulated entities move in the geographic area of the scenario, their update and subscription rectangles will correspondingly move in the DDM coordinate space. The movement of the update and subscription regions can result in regions that previously were intersecting to become non-intersecting and regions that were previously non-intersecting to become intersecting. The changes to region intersections imply changes to the desired inter-federate data flow. Thus, to properly route messages between federates, the RTI must respond to the entities' movement by repeatedly solving a computational geometry problem: given a set of axis-parallel hyper-rectangles in a multidimensional coordinate space, find all intersecting pairs of rectangles. In HLA DDM terminology, this process is known as "matching."

2.2 SAF Systems

Military simulation systems often include simulated entities (such as tanks, aircraft, or individual humans), which are generated and controlled by computer software rather than by human crews or operators [3]. The constructive entity-based combat models that generate and control such entities are known as semi-automated

forces (SAF) systems; “automated” because software generates entities’ behavior automatically and “semi-” because the entities behavior may be monitored and optionally controlled by a human operator.

In a military training application, SAF systems are often used to generate opponents against which human trainees engage in virtual battles. Doing so with a SAF system is preferable to having additional human crews in simulators control the hostile forces because SAF systems are both less expensive, as they reduce the need for a large number of simulators not available for the trainees, and more flexible, in that they can be configured to use the tactical doctrine of a particular adversary more readily than retraining human opponents. SAF systems can also generate friendly forces, allowing a small group of trainees to practice teamwork within a large friendly force. In non-training applications, such as testing a revised tactical doctrine or assessing the effect of an enhanced weapon, SAF systems typically are used to generate all of the entities involved in the simulation, allowing the analysis scenarios to be executed repeatedly to support statistical analysis without exhausting human operators.

The entities generated and controlled by the SAF system exist in a battlefield that is a simulated subset of the real world, so the physical events and phenomena on the battlefield, such as movement and combat, must be modeled within the SAF system. In addition, SAF systems generate the behavior of the entities they control that allows those entities to react autonomously to the battlefield situation as represented in the simulation [16]. The behavior must be both behaviorally realistic, in that it appears to be similar to human behavior in the same situation [23], and doctrinally consistent, in that the actions of the SAF-controlled entities should be consistent with doctrine of the entities the SAF is simulating.

3 Research Question and Procedure

This section formally states the research question addressed and describes the procedure and software used to investigate it.

3.1 Research Question and Procedure

Since HLA was introduced, much research effort has been devoted to developing data structures and algorithms to perform DDM matching as efficiently as possible, for example, [1, 8, 12, 13, 15, 22]. In this study, a different approach was taken, inspired by earlier work intended to reduce the computational cost of intervisibility determinations in SAF systems [20]. Instead of (or perhaps in addition to) performing DDM matching as *efficiently* as possible, the approach is to perform DDM matching as *seldom* as possible, as long as the results of the federation’s simulation are not compromised. In other words, this study addresses the question

of how infrequently DDM matching might be performed before the results of the simulation are negatively affected.

The effects of reducing DDM matching frequency could have been investigated by modifying a production military SAF system, such as OneSAF [9]. However, such SAF systems are extremely large and complex bodies of software, require specialized skills to operate, and may contain sensitive or classified information, making them problematic for laboratory experiments. Instead, a custom SAF system named RSAF, similar to production military SAF systems but abstracted and simplified, was developed and verified. RSAF was then used to experimentally assess the effect of performing DDM matching at various frequencies.

Hostile battlefield entities can fire at each other only after sighting each other, and a sighting requires an exchange of messages between the federates simulating the entities, and an exchange of messages requires that the entities' DDM rectangles intersect. Therefore, the effect of reducing the frequency of DDM matching was measured by tracking three times for each pair of hostile battlefield entities: when their DDM rectangles first intersected, when they first sighted each other, and when they first shot at each other. These times were recorded during simulation execution by the custom SAF for later analysis.

3.2 RSAF

RSAF is an abstract constructive entity-level SAF system, implemented entirely in the R language [21] (hence the name "RSAF"). The many complex features of full-function SAF systems that are not relevant to the goals of this study are not implemented in RSAF. For example, RSAF has no operator interface because it executes without operator intervention, it has no tactical behavior generation because RSAF entities move along scenario-specific preplanned paths and use simple target selection rules, and it has no interoperability protocol interface because it operates only as a standalone model. However, RSAF does include both the features of a SAF system and the features of an HLA federation that are necessary to the study. Regarding the HLA features, RSAF performs the DDM matching function, which is usually done by the RTI rather than an individual federate, and it tracks the number of network messages that would have been sent and received with and without DDM operating in the federation.

RSAF represents the effect of terrain on line of sight in an abstract way. To determine if two entities can sight each other, and thus potentially engage in direct fire combat, conventional SAF systems perform a computationally expensive check of whether a line segment in three-dimensional space connecting the two entities intersects any of the polygons that make up the terrain's surface, or its features, such as buildings or trees. In contrast, RSAF determines whether a line of sight exists between two entities stochastically, by comparing a random number to an intervisibility probability. That probability is a function of terrain density (sighting is more likely in open terrain, e.g., desert, than it is in closed terrain, e.g., urban), of

range (sighting is more likely when entities are closer), and of the sighting entities' sensor capabilities (sighting is more likely with thermal sights than with visual scanning) [5].

Similarly, determining whether an entity's direct fire shot hits its target and determining whether a hit destroys the target are also done stochastically. (Stochastic resolution of intervisibility and combat was previously used to good effect in an experimental combat model developed for the Defense Advanced Research Projects Agency [18]). Equations (1), (2), and (3) are the formulas used for the stochastic sighting, hit, and kill determinations. They use the exponential distribution probability density function $f(x) = \lambda e^{-\lambda x}$ with rate $\lambda = 1$ and parameter x a function of range, terrain density, and the entity's sensors, weapon accuracy, and weapon lethality; the latter three are entity-specific constants and are denoted as P_s , P_h , and P_k in the equations.

$$P(\text{sighting}) = e^{-\left(\frac{\text{range}}{P_s} \bullet \text{terrain density}\right)} \quad (1)$$

$$P(\text{hit}) = e^{-\left(\frac{\text{range}}{P_h} \bullet \text{terrain density}\right)} \quad (2)$$

$$P(\text{kill}) = e^{-\left(\frac{\text{range}}{P_k} \bullet \text{terrain density}\right)} \quad (3)$$

RSAF executes in time steps, each representing 0.2 seconds of time. Each time step has four phases: movement, matching, sighting, and shooting. In the movement phase, each active (non-destroyed) entity has its location updated by moving it along its assigned path at its specified speed. In the matching phase, the entities' update and subscription rectangles are repositioned at the entities' updated locations. Then, each active entity's subscription rectangle is compared to every other active entity's update rectangle to determine if the subscription and update rectangles intersect. If they do intersect, a network message containing an updated entity location is counted as being sent from the entity with update rectangle to the entity with the subscription rectangle. In the sighting phase, for each pair of active entities that have intersecting rectangles that have not already sighted each other, a stochastic intervisibility determination is performed as described earlier. Finally, in the shooting phase, for each active entity that has one or more sighted active enemy entities and for which an entity-specific minimum firing interval has passed since the entity last fired, a stochastic hit determination is performed. If a hit is achieved, a stochastic kill determination is performed, possibly changing the target entity's status from active to destroyed.

RSAF optionally produces a graphical map view of the state of an executing scenario. Figures 2 and 3 show RSAF executing the "Hasty Attack" scenario (described later) at the beginning of (Fig. 2) and approximately 9.8 simulated minutes into (Fig. 3) the 15-minute scenario. In the figures, the circle and diamond symbols represent entities; circles are blue entities and diamonds are red. The chains

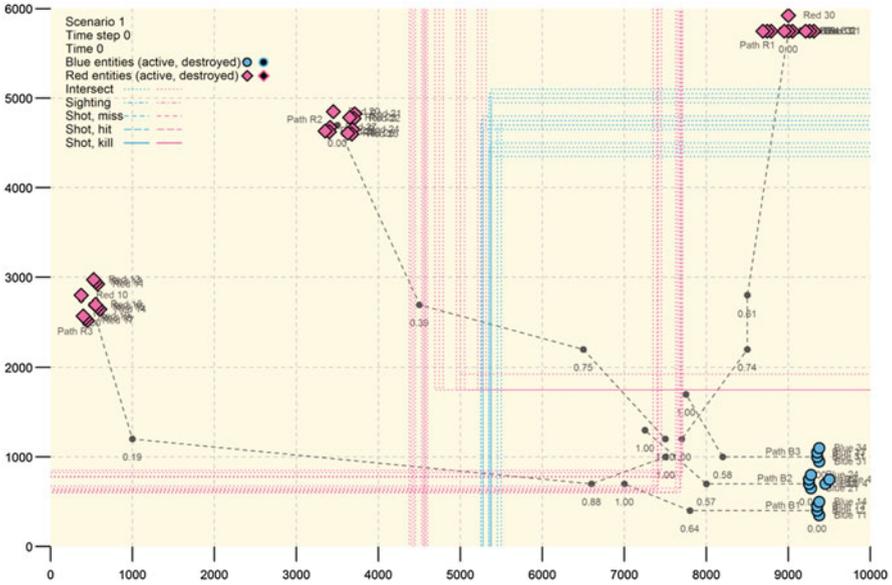


Fig. 2 RAAF map view, showing the initial state of the Hasty Attack scenario

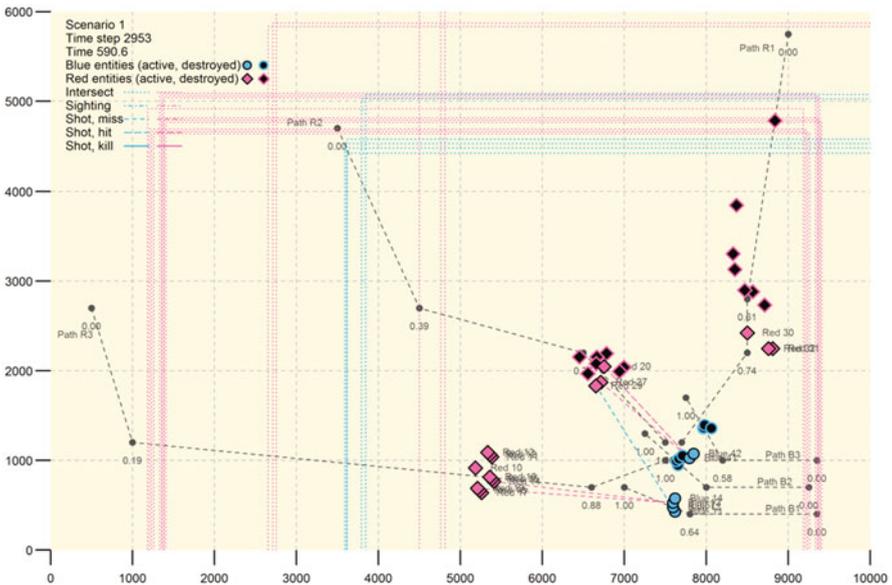


Fig. 3 RAAF map view, showing an execution of the Hasty Attack scenario at 9.8 minutes

of dashed segments are the paths the entities follow. The large dashed squares centered on the active entities are their subscription rectangles; the update rectangles are much smaller and are not shown. The segments connecting some entity symbols represent the inter-entity actions that occurred during the current time step, with different line types signifying different action types: intersection, sighting, shot that missed the target, shot that hit but did not kill the target, and shot that killed the target (the connecting lines are color coded to indicate the acting entity). In Fig. 3, both red and blue have suffered substantial losses; 7 of the 14 blue vehicles and 14 of the 30 red vehicles have been destroyed.

RSAF is initialized with a simulation scenario to execute. The initialization data for a scenario include the entities in the scenario, with performance data such as movement speed and sensor effectiveness for each, the paths the entities follow during execution, terrain parameters, including terrain density and extent, and scenario execution parameters, such as scenario duration and whether or not to simulate shooting.

Because of its abstract representations, RSAF executes approximately $15\times$ faster than real time on standard office workstations, in spite of the fact that its implementation language R is interpreted.

A simulation model should be subject to “credibility evaluation” before using its results [6]. The terms verification, validation, and accreditation denote different objectives in the overall process of assessing a model’s correctness and usability [17]. The RSAF code includes verification tests that check the logical consistency of its results. For example, for each pair of hostile entities, the pair’s sighting time must be greater than or equal to its first intersect time, and its first shot time must be greater than or equal to its first sighting time. A total of 59 distinct verification tests are included in the RSAF code and are executed automatically each time RSAF is run. If one or more of them fail, the execution stops, the results discarded, and the problem resolved. No verification tests failed during any of the RSAF executions used to collect the data reported in this chapter.

4 Results, Conclusions, and Future Work

This section describes the test scenario, reports the results of the experiments, presents the conclusions drawn from those results, and suggests possible future work related to DDM matching frequency and RSAF.

4.1 Scenario

The “Hasty Attack” scenario illustrated in Figs. 2 and 3 was used for this study. It is based on Cold War-era Soviet and US tactical doctrine and force structures and was previously used for research into algorithms for SAF intervisibility and

behavior generation [10]. In the “Hasty Attack” scenario, three red companies, each equipped with ten armored vehicles organized into three platoons of three vehicles and one command vehicle, are conducting an offensive action without extensive preparation or maneuver in an attempt to displace blue forces from a key terrain area before blue can deploy defensively. Simultaneously, a blue company, consisting of three platoons of four vehicles and two command vehicles, is seeking to rapidly deploy from a maneuver formation into an organized defensive position. The three red companies begin the scenario at the top right, top center, and left center and the blue company begins at bottom right of the 10 kilometer \times 6 kilometer terrain area. The scenario executes for 900 seconds, that is, 15 minutes, of simulated time.

4.2 Results

The “Hasty Attack” scenario was implemented and tested. It was then executed 30 times at each of eight different matching frequencies in each of two modes, shooting enabled and shooting disabled, for a total of $2 \cdot 8 \cdot 30 = 480$ trials.

Figures 4 and 5 show the results of executing the test scenario 30 times at eight different matching intervals with shooting disabled. In the figures, a matching interval value of k means that matching was performed once every k time steps; for example, at matching interval = 1, matching is performed every time step, and at matching interval = 128, matching is performed once every 128 time steps. Figure 4 shows the mean time of first DDM rectangle intersection and the mean time of first sighting of a hostile entity for all entities in the scenario. (Entities that did not have a first intersection or first sighting are not included in the respective means). As expected, the mean first intersection time increases monotonically as the matching interval increases; if matching is done less frequently, the first intersection will be found later. Similarly, the mean first sighting time also generally increases as matching interval increases; the increase is not quite strictly monotonic (e.g., mean first sighting time is earlier at matching interval = 2 than it is at matching interval = 1) because sighting is resolved stochastically. Figure 4 shows how much the mean first intersection time and the first sighting time were delayed, with respect to the time for matching interval = 1, at each of the matching intervals tested. In other words, the figure shows the effect of reducing the frequency of matching.

Figures 6 and 7 show the results of executing the test scenario 30 times at eight different matching intervals with shooting enabled. (Of course, this is the more typical setting for combat simulation). Figure 6 shows the mean time of first intersection, the mean time of first sighting, and the mean time of first shot for all entities in the scenario. (As before, entities that did not have a first intersection, first sighting, or first shot are not included in the respective means). As expected, all three mean times increase as the matching interval increases; none of the increases are strictly monotonic because of the random effects of both stochastic sighting and stochastic combat. For example, combat may destroy an entity, that, had it survived longer, might have had a first intersection and a first sighting with a more distant

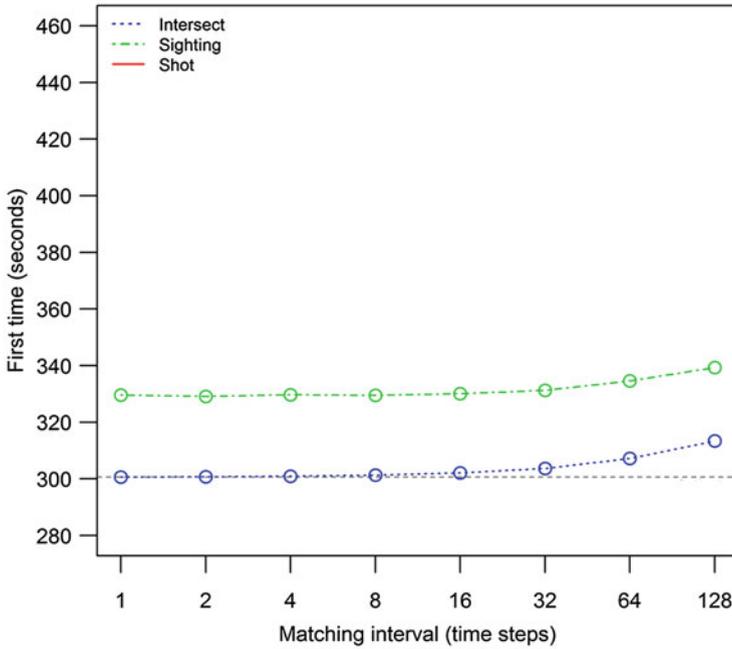


Fig. 4 Mean first intersect and sighting times, without shooting

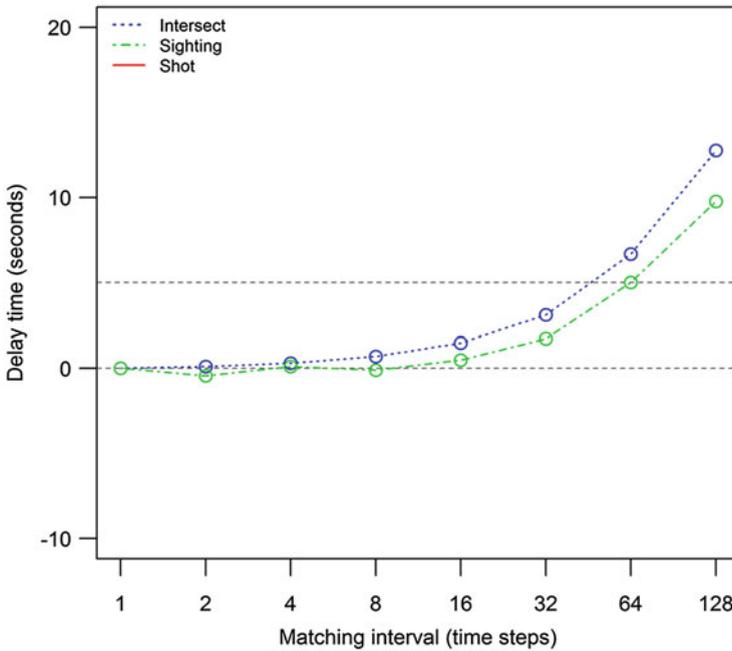


Fig. 5 Mean first intersect and sighting delays, without shooting

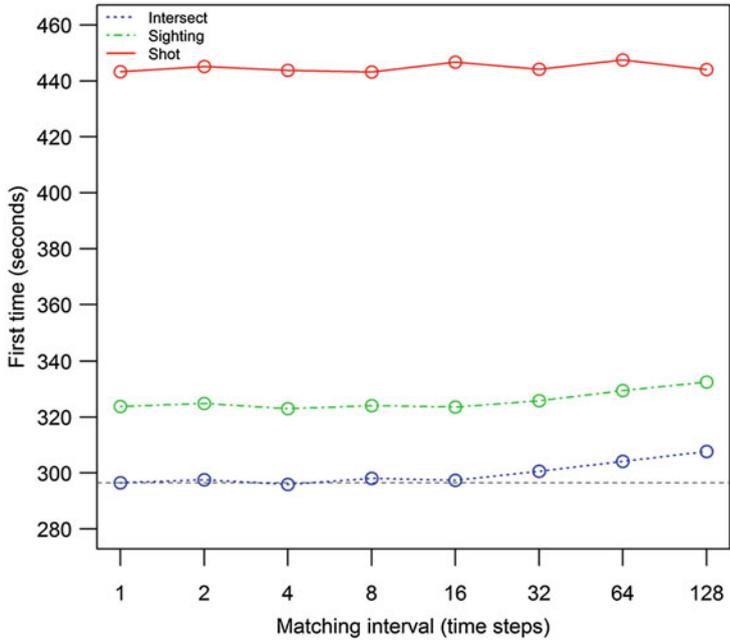


Fig. 6 Mean first intersect, sighting, and shot times, with shooting

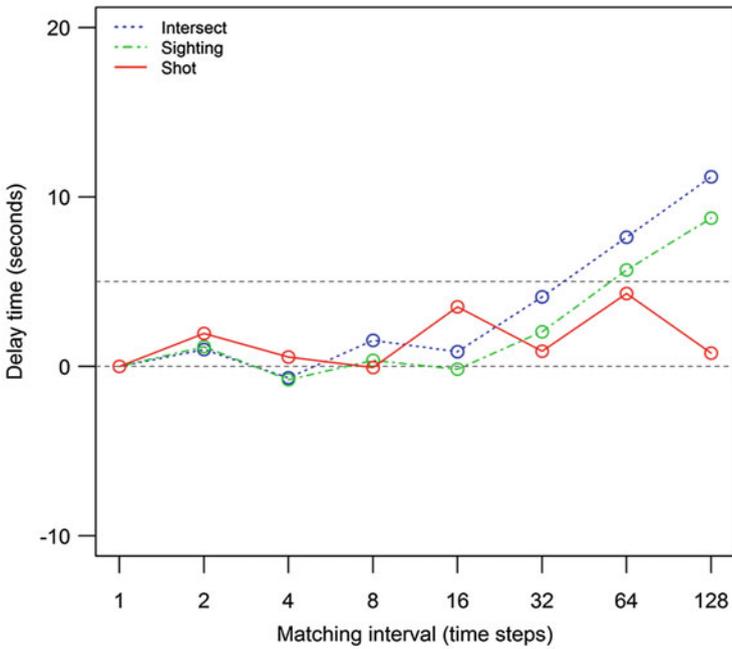


Fig. 7 Mean first intersect, sighting, and shot delays, with shooting

Table 1 Mean first times and delays for intersect, sighting, and shot, averaged over 30 trials, for each shooting mode and matching interval

Shooting mode	Matching interval	Intersect		Sighting		Shot	
		First	Delay	First	Delay	First	Delay
Disabled	1	300.59	0.00	329.58	0.00	n.a.	n.a.
	2	300.69	0.10	329.14	-0.44	n.a.	n.a.
	4	300.89	0.30	329.69	0.11	n.a.	n.a.
	8	301.28	0.69	329.47	-0.11	n.a.	n.a.
	16	302.06	1.47	330.05	0.47	n.a.	n.a.
	32	303.70	3.11	331.29	1.71	n.a.	n.a.
	64	307.26	6.67	334.60	5.02	n.a.	n.a.
	128	313.36	12.77	339.36	9.78	n.a.	n.a.
Enabled	1	296.55	0.00	323.77	0.00	443.25	0.00
	2	297.56	1.01	324.92	1.15	445.20	1.95
	4	295.88	-0.67	322.99	-0.78	443.81	0.56
	8	298.10	1.55	324.13	0.36	443.18	-0.07
	16	297.43	0.88	323.61	-0.16	446.77	3.52
	32	300.67	4.12	325.82	2.05	444.16	0.91
	64	304.17	7.62	329.47	5.70	447.56	4.31
	128	307.75	11.20	332.52	8.75	444.04	0.79

entity at a later time thereby increase the trial’s mean first intersection and first sighting time. Figure 6 shows how much the mean first intersection time and the first sighting time were delayed, with respect to the time for matching interval = 1, at each of the matching intervals tested. In other words, the figure shows the effect of reducing the frequency of matching.

Table 1 reports the numeric values for first times and delays for each of the two shooting modes (disabled and enabled), eight matching intervals (1, 2, 4, 8, 16, 32, and 64), and three actions (intersect, sighting, and shot). The times shown in Table 1 are the times plotted in Figs. 4–7.

4.3 Conclusions

As expected, larger matching intervals produce larger delays in the mean times of first intersection, first sighting, and first shooting. Intuitively, larger delays are undesirable. But how much delay can be tolerated without significantly affecting a scenario’s outcome? To approach this question, refer again to Fig. 7, which shows mean delay times with shooting enabled (the normal mode of execution for combat simulations). The figure has two horizontal reference lines, one at delay time = 0 second (i.e., no delay), which is the delay at matching interval = 1, and one at delay time = 5 seconds. The latter delay time (5 seconds) is tactically relevant because it is the approximate time required for a competent US Army tank crew to

reload, aim, and fire their tank's main gun once (assuming a target has been sighted). Arguably, mean delays less than 5 seconds will not reduce the mean number of shots taken and thus may have little effect on the scenario outcome in a tactical sense. Figure 7, and the detailed data that it depicts, suggests that matching intervals of up to 32 can be used without exceeding the 5-second threshold.

These results suggest the possibility of significantly reducing the computational expense of DDM matching. If each matching operation is assumed to require approximately the same computation, performing matching once every 32 time steps, instead of every time step, could save as much as $31/32 \approx 97\%$ of the computation required for matching. (In practical implementations, the savings may be less due to unavoidable computational overhead). Of course, HLA federations may differ in characteristics that affect the computational savings realized, including number of entities, how quickly the entities move relative to the sizes of their subscription and update rectangles, and how frequently the entities interact. Consequently, simply recommending a matching interval of 32 time steps for all federations is not justified. However, it is recommended that HLA federation implementers experiment with reducing the frequency of DDM matching so as to reduce the computational expense of matching and thereby increase the scalability of their federations.

4.4 Future Work

RSAF and the study reported in this chapter could be extended in several ways, including the following:

- Both of RSAF's HLA federation functions (DDM matching and network message counting) assume that each RSAF entity is being simulated by a separate federate. Enhancing RSAF to model multiple entities per notional federate would support additional studies.
- Additional SAF-specific features could be added to RSAF, including elementary tactical behaviors (e.g., move away from hostile entities once losses pass a self-preservation threshold) and a less abstract representation of terrain.
- This study measured only mean first times and mean delay times of intersection, sighting, and shooting. Additional response variables could be considered, including scenario outcomes (e.g., red and blue losses) and the proportion of entities never sighted.
- Additional test scenarios, with initial entity positions and movement paths different from the "Hasty Attack" scenario, could be implemented and evaluated.

References

1. J. Ahn, C. Sung, T.G. Kim, A binary partition-based matching algorithm for data distribution management in a high-level architecture-based distributed simulation. *Simulation* **88**(11), 1350–1367 (2012)
2. P. Bocciarelli, A. D’Ambrogio, A. Falcone, A. Garro, A. Giglio, A model-driven approach to enable the simulation of complex systems on distributed architectures. *Simulation* **95**(12), 1185–1211 (2019)
3. A.J. Collins, M.D. Petty, D. Vernon-Bido, S. Sherfey, A call to arms: Standards for agent-based modeling and simulation. *J. Artif. Soc. Soc. Simul.* **18**(3) (2015). <https://doi.org/10.18564/jasss.2838>
4. J.S. Dahmann, F. Kuhl, R. Weatherly, Standards for simulation: As simple as possible but not simpler: The high level architecture for simulation. *Simulation* **71**(6), 378–387 (1998)
5. K.L. Foster, M.D. Petty, An examination of the tactical impact of drone swarms using a semi-automated forces system calibrated to the historical results of the battle of 73 easting. In: *Proceedings of the 2019 AlaSim Conference and Exhibition*, Huntsville, AL, 24 Oct 2019
6. Y. Laili, L. Zhang, Y. Luo, A pattern-based validation method for the credibility evaluation of simulation models. *Simulation* **96**(2), 151–167 (2020). <https://doi.org/10.1177/0037549719856100>
7. K.L. Morse, M. Zyda, Multicast grouping for data distribution management. *Simul. Pract. Theory* **9**(3–5), 121–141 (2002)
8. K. Pan, S.J. Turner, W. Cai, Z. Li, A dynamic sort-based DDM matching algorithm for HLA applications. *ACM Transact. Model. Comput. Simul.* **21**(3), 1–17 (2011). <https://doi.org/10.1145/1921598.1921601>
9. Parsons D, Surdu J, Jordan B, OneSAF: A next generation simulation modeling the contemporary operating environment. In: *Proceedings of the 2005 European Simulation Interoperability Workshop*, Toulouse, France, 27–29 June 2005
10. M.D. Petty, Computational Geometry Techniques for Terrain Reasoning and Data Distribution Problems in Distributed Battlefield Simulation. Dissertation, University of Central Florida (1997)
11. M.D. Petty, P.S. Windyga, A high level architecture-based medical simulation. *Simulation* **73**(5), 279–285 (1999)
12. M.D. Petty, Geometric and algorithmic results regarding HLA data distribution management matching. In: *Proceedings of the Fall 2000 Simulation Interoperability Workshop*, Orlando, FL, 17–22 Sept 2000
13. M.D. Petty, K.L. Morse, Computational complexity of HLA data distribution management. In: *Proceedings of the Fall 2000 Simulation Interoperability Workshop*, Orlando FL, 17–22 Sept 2000
14. M.D. Petty, Comparing high level architecture data distribution management specifications 1.3 and 1516. *Simul. Pract. Theory* **9**(3–5), 95–119 (2002)
15. M.D. Petty, K.L. Morse, The computational complexity of the high level architecture data distribution management matching and connecting processes. *Simul. Model. Pract. Theory* **12**(3–4), 217–237 (2004)
16. M.D. Petty, Behavior generation in semi-automated forces, in *The PSI Handbook of Virtual Environment Training and Education: Developments for the Military and Beyond*, VE Components and Training Technologies, ed. by D. Nicholson, D. Schmorrow, J. Cohn, vol. 2, (Praeger Security International, Westport, 2009), pp. 189–204
17. M.D. Petty, Verification, validation, and accreditation, in *Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains*, ed. by J. A. Sokolowski, C. M. Banks, (Wiley, Hoboken, 2010), pp. 325–372
18. M.D. Petty, J. Panagos, J.P. Joseph, R.W. Franceschini, Validation using comparison testing of three constructive combat models. In: *Proceedings of the Fall 2011 Simulation Interoperability Workshop*, Orlando, FL, 19–23 Sept 2011

19. M.D. Petty, J. Kim, S.E. Barbosa, J. Pyun, Software frameworks for model composition. *Model. Simul. Eng.* (2014). <https://doi.org/10.1155/2014/492737>
20. S. Rajput, C.R. Karr, M.D. Petty, M.A. Craft, Intervisibility heuristics for computer-generated forces, in *Distributed Interactive Simulation Systems for Simulation and Training in the Aerospace Environment*, ed. by T. L. Clarke, (SPIE Press, Bellingham, 1995), pp. 299–327
21. R Core Team, *R: A language and environment for statistical computing*. In: R Foundation for Statistical Computing. (2019), <https://www.R-project.org/>. Accessed 30 Dec 2019
22. C. Raczy, G. Tan, J. Yu, A sort-based DDM matching algorithm for HLA applications. *ACM Transact. Model. Comput. Simul.* **15**(1), 14–38 (2005). <https://doi.org/10.1145/1044322.1044324>
23. B.G. Silverman, G. Bharathy, N. Weyer, What is a good pattern of life model? Guidance for simulations. *Simulation* **95**(8), 693–706 (2018)

Research on Repair Strategy of Heterogeneous Combat Network



Yanyan Chen, Yonggang Li, Shangwei Luo, and Zhizhong Zhang

1 Introduction

With the development of weapons and equipment systems, most of the weapons in modern military operations are capable of reconnaissance, decision-making, attacking, and so on. Coordination and cooperation between operational entities is emphasized in the information warfare network. In order to facilitate the research, the combat entity and their information interaction are usually abstracted into a heterogeneous combat network with multiple types of nodes and edges [1, 2]. At present, many achievements have been made in the research of heterogeneous combat networks. References [3, 4] analyze heterogeneous combat networks based on topological characteristics and propose indicators to measure system capabilities. The analysis method based on complex networks can analyze the complex characteristics of combat system [5].

In the course of combat, the combat entity is easy to be attacked by the enemy, resulting in network damage. Such network damage usually cannot be repaired in time. When a functional node is damaged, the edges connected to the node cannot be connected, and the combat capability of the combat network decreases accordingly. The research on combat network repair has become a hot topic.

According to the difference and connection between topological information and communication transmission network, Miao analyzed the relationship between topological theory and network survivability and introduced the practical application of topological algorithm in network repair [6]. By analyzing different attack strategies, the repair model of complex network is established [7–10]. Reference [11] describes the network repair problem of the attacked nodes in the

Y. Chen (✉) · Y. Li · S. Luo · Z. Zhang
Chongqing University of Posts and Telecommunications, Chongqing, China

military communication network, and the topology of the communication network is repaired by adding edges to the network. Jiang et al. proposed a scheme to add a small number of links to the existing network topology to improve network performance and proposed four effective edge addition strategies [12]. However, all these studies do not take into account the characteristics of system and also ignore the complexity problem of the repair algorithm in the large-scale network.

In the recent repair strategies, most model and analysis methods only explore traditional networks. In other words, in current research, the entities and the interaction between entities are modeled as the same kind of nodes or edges when the combat entities are networked. These studies do not consider the heterogeneity of nodes and edges, so these repair algorithms cannot be applied to heterogeneous combat networks. In this chapter, a repair model is established based on heterogeneous combat network (HCNR, A Heterogeneous Combat Network Repair Algorithm). Four attack methods are used for simulation. The results show that the proposed algorithm is effective and suitable for networks of different sizes.

This chapter is structured as follows. Related works on combat network modeling and analyses are summarized in Sect. 2. In Sect. 3, an HCNR model is established. Section 4 provides simulation experiment and analysis. Finally, conclusions are discussed in Sect. 5.

2 Modeling and Analysis for Heterogeneous Combat Network

2.1 Modeling

When the combat system is modeled as a heterogeneous network, the function of the combat entity is modeled as a corresponding functional node. Functional nodes can be divided into sensor node (S), decider node (D), and influence node (I). The edges between the three types of functional nodes represent different information interactions. The rules are as follows:

- $S \rightarrow D$ and $D \rightarrow I$: Every S and I has at least one edge connected to D . Both S and I can be connected to several D . These edges represent the information transfer between S or I and D .
- $S \rightarrow S$: There can be edges between different S . This edge represents the degree of information sharing between the sensor nodes.
- $D \rightarrow D$: There can be edges between different D . This edge represents the connectivity between decider nodes.
- The edges in the heterogeneous network are directional edges, and there are no directly connected edges from S to I .

The combat behavior is abstracted into the combat flow, and the combat capability is measured by the functional chain based on the structural attribute of the

combat system [13, 14]. A functional chain is defined as a link from S to I through D , as shown in Fig. 1. In modern warfare, operational entities are closely connected, and different functional chains interweave to form heterogeneous combat networks, as shown in Fig. 2.

2.2 Functional Reliability

In the heterogeneous combat networks, network combat efficiency can be represented by the shortest functional chain length between all S and I [15]. The shorter the length, the higher the operational efficiency. The shortest distance matrix B between node S and I is calculated from the adjacency matrix A .

$$B = \begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ d_{m1} & d_{m2} & \cdots & d_{mn} \end{pmatrix} \tag{1}$$

The network combat efficiency is:

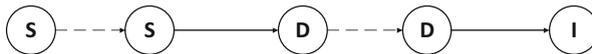
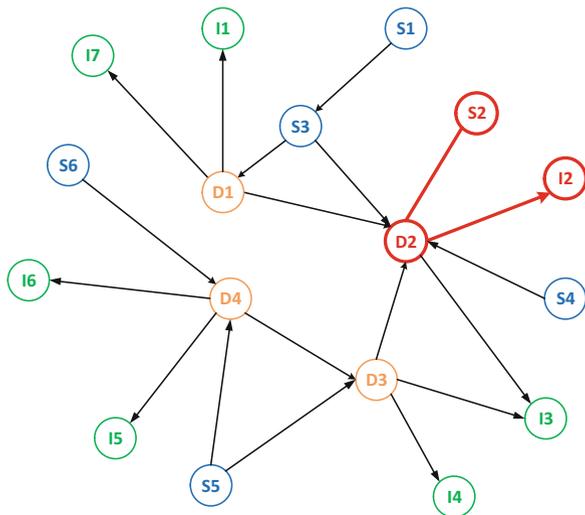


Fig. 1 Schematic of functional chain model

Fig. 2 The functional chain in a heterogeneous combat network



$$Q_G = \sum \frac{1}{d_{ij}} \tag{2}$$

where $d_{ij} \in B$ and $i \in S, j \in I$.

The number of functional chains in a combat network can indicate network robustness [16, 17]. The more the number of functional chains, the higher the redundancy and robustness of network operations. For the calculation of the number of functional chains in combat network, a robust index algorithm based on adjacency matrix is adopted. Since I is only connected to D , the number of paths calculated using the adjacency matrix is the total number of functional chains.

For the adjacency matrix A , after A^k calculation, the matrix element of the i th row and the j th column obtained represents the number of paths that can be reached through k steps from node i to node j .

$$A^k = \begin{pmatrix} a_{11}^{(k)} & a_{12}^{(k)} & \cdots & a_{1n}^{(k)} \\ a_{21}^{(k)} & a_{22}^{(k)} & \cdots & a_{2n}^{(k)} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1}^{(k)} & a_{m2}^{(k)} & \cdots & a_{mn}^{(k)} \end{pmatrix} \tag{3}$$

When the length of the functional chain becomes longer, the operational efficiency of the functional chain will decrease. By weighting the combat capability of the functional chain, the influence of the longer functional chain on the combat effect is reduced, that is, the longer the functional chain, the less weight is given. The weighted functional chain combat capability can be expressed as follows:

$$S_{ij} = \sum_{k=2}^{\infty} \frac{1}{k} \times a_{ij}^{(k)} \tag{4}$$

where $a_{ij}^{(k)} \in A^k$ and $i \in S, j \in I$.

The network robust index is:

$$S_G = \sum_{i \in S} \sum_{j \in I} S_{ij} \tag{5}$$

In this chapter, functional reliability is used to measure network combat capability, which is defined as the weighted sum of network operational efficiency and network robustness index, that is,

$$F_G = \alpha Q_G + \beta S_G \tag{6}$$

where α and β are the weight coefficients.

3 Repair Model Framework

3.1 Design of Repair Model

In recent research, the repair strategy studied is to add edges between the remaining functional nodes. When the scale of combat network is increased, the operation speed is slow and the solution result is not ideal. Aiming at these problems, this chapter analyzes the nature of heterogeneous combat network and designs a repair model suitable for different scale combat network.

3.1.1 Optional Edge

Optional edges refer to the feasible solution of the HCNR, that is, the edges that can be added in the damaged network. With the increase of network size, the more nodes, the more optional edges, and the higher the complexity of the repair algorithm. According to the characteristics of heterogeneous networks, when selecting optional edges, this chapter first selects some nodes (called feasible nodes) that meet the selection conditions and then takes the edges that meet the selection conditions between these nodes as optional edges.

The degree of a node represents its degree of association with other nodes. The indegree of sensor node S represents the degree of information sharing between this node and other sensor nodes. The degree of decider node D represents its closeness with other decider nodes. The degree of influence node I represents its mission load. According to the degree of the node, the selection conditions of different functional nodes are also different, as follows:

- The sensor node S : The indegree of the sensor node S is arranged from large to small, and the first n_L nodes are selected.
- The decider node D : The degree of the decider node D is arranged from large to small, and the first n_L nodes are selected.
- The influence node I : The degree of the influence node I is arranged from small to large, and the first n_L nodes are selected.

n_L is the maximum number of edges that the network can increase, which is proportional to the size of network nodes, that is $n_L = \varepsilon \times N$, and ε is the constant coefficient, and N is number of network nodes.

The edges of heterogeneous combat network are directed edges, and the types of edges that the network can add include $S \rightarrow S$, $S \rightarrow D$, $D \rightarrow D$, $D \rightarrow I$. An edge that satisfies the edge adds type between the unconnected feasible points as an optional edge to the HCNR.

3.1.2 Repair Model

The HCNR is designed from two aspects: one is the objective function and the other is the constraint conditions on the added edges.

Operational capability can be measured by the functional reliability of combat network. In this chapter, combat network functional reliability is used as the objective function and maximization of functional reliability is taken as the objective of the HCNR.

$$\max F_G = \max (\alpha Q_G + \beta S_G) \tag{7}$$

When topology repairs are made, the edges to be added are constrained, rather than the edges between all unconnected nodes.

1. Optional Edge Constraint

In the HCNR based on the characteristics of heterogeneous combat network, when adding edges to repair, the added edges should satisfy the constraint condition of optional edges. That is, the solution of the HCNR should be optional edges.

2. Cost Constraint

Connection costs should be considered when establishing connections between functional nodes. The reliability and cost of communication between functional nodes are related to the distance between operational entities. Therefore, the HCNR uses the distance between operational entities to measure the cost of establishing the edge, that is,

$$b_{ij} = \alpha \times d_{ij} \tag{8}$$

$$B = \sum b_{ij} \tag{9}$$

where d_{ij} is the distance between the i th combat entity and the j th combat entity, and α is the distance coefficient. This chapter defines the upper limit of the cost of adding edges to the network as B_{\max} .

The HCNR is as follows:

$$\begin{aligned} &\max F_G \\ &s.t \begin{cases} \Delta m_{ij} \in \{0, 1\}, & i \in S_k, j \in S_k, D_k \text{ or } i \in D_k, j \in D_k, I_k \\ \Delta m_{ij} \neq \Delta m_{ji} \\ B \leq B_{\max} \end{cases} \end{aligned} \tag{10}$$

where $S_k, D_k,$ and I_k are, respectively, feasible point sets of three node types.

3.2 Solution of Repair Model

In this chapter, the artificial bee colony algorithm is used to solve the HCNR and the optimal edge-adding scheme is obtained [18].

The solving steps of the artificial bee colony algorithm are (1) selecting the optional edges, then encoding the parameters; (2) initializing of population and calculating of fitness; (3) leader stage; (4) follower stage; (5) scouter stage; (6) judging whether the end condition is reached. If so, output the optimal solution; otherwise, return to the (3) and continue to execute the algorithm.

3.2.1 Parameter Coding

The edges that meet the conditions are coded, as shown in Table 1. Edge sequence $Y_k = (i, j)$ means that node i points to the edge of node j , and the corresponding encoding value x_k is 0 or 1. $x_k = 1$ means that the corresponding edge is established in the damaged network, and $x_k = 0$ means that it is not established. The encoded value of the permutation is an optimization variable $X_i = \{x_1, x_2, \dots, x_n\}$.

3.2.2 Population Initialization

The individuals in the initial population are randomly generated. Different from the traditional algorithm, when initializing in the HCNR, the number of encoding values equal to 1 in each individual is no more than n_L and the total cost of each solution is no more than B_{max} . The algorithm generates random solutions that satisfy the initial conditions. The set of solutions is called $Z_X = \{X_1, X_2, \dots, X_{SN}\}$.

3.2.3 Fitness Function

It is hoped that the network can have higher functional reliability after repair. Therefore, functional reliability is used to measure fitness. In the HCNR, the fitness is defined as the normalized functional reliability, that is, the ratio between the functional reliability F_{G_i} of the network corresponding to each repair scheme X_i and the functional reliability F_G of the original network. The fitness function is expressed as follows:

$$Fit_i = \frac{F_{G_i}}{F_G} \tag{11}$$

Table 1 Parameter coding rule

Edge sequence	Y_1	Y_2	...	Y_k	...	Y_n
Encoded value	x_1	x_2	...	x_k	...	x_n

4 Simulation Results and Analysis

4.1 Simulation Environment Settings

The heterogeneous combat network is shown in Fig. 3. The blue node is the sensor node S , the red node is the decider node D , and the yellow node is the influence node I . The blue, green, red, and yellow edges are shown $S \rightarrow S$, $S \rightarrow D$, $D \rightarrow D$, $D \rightarrow I$.

This chapter adopts the network damage scheme of attacking nodes. For the heterogeneous combat network, there are four attack strategies, respectively, sensor node attacking strategy, decider node attacking strategy, influential nodes attacking strategy, and random-node attacking strategy.

The strength coefficient of attack node f_N refers to the ratio between the number of attack nodes and the total number of network nodes, that is,

$$f_N = \frac{m_N}{M_N} \tag{12}$$

where N refers to the attack strategy, m_N is the number of attack nodes of the N th class, and M_N is the total number of nodes of the N th class.

The influence of different values of f_N on functional reliability and the results of different node attack strategies are shown in Fig. 4. With the increase of f_N , the number of nodes in the damaged network is decreasing, and the functional reliability is also decreasing. When f_N is equal, in the random attack strategy, the number of attacked nodes is greater, and the functional reliability of network decreases relatively quickly. Decider node plays a key role in the combat network, so attacking decider node D is more damaging to the combat network than attacking sensor node S and influence node I .

The improvement rate was used to evaluate the performance of the repair algorithm. In this chapter, the improvement rate refers to the ratio of the improved

Fig. 3 The heterogeneous combat network model

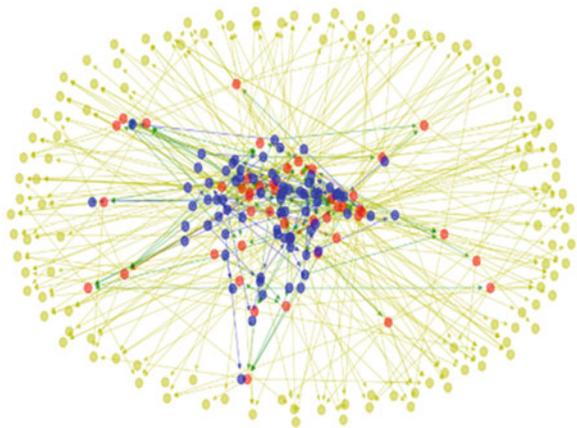
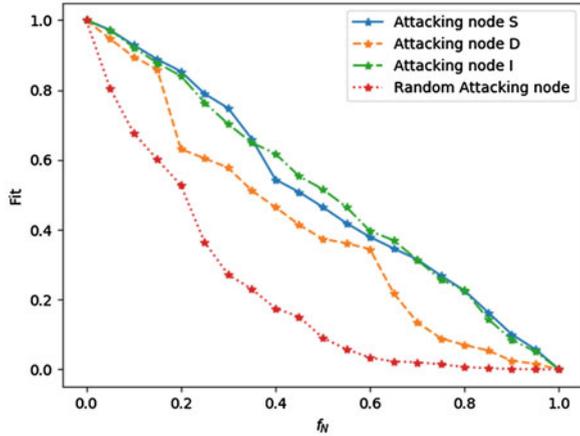


Fig. 4 The function of functional reliability and node attack strength coefficient under four attack strategies



functional reliability after repair to the functional reliability of the damaged network before repair, that is,

$$C = \frac{\Delta F}{F_{G_i}} = \frac{F'_{G_i} - F_{G_i}}{F_{G_i}} \tag{13}$$

where F_{G_i} is the functional reliability of the damaged network, and F'_{G_i} is the functional reliability of the damaged network after repair.

4.2 Results and Analysis

1. Effectiveness Analysis of HCNR

In the simulation network, there are a total of 240 nodes, among which the number of S , D , and I is 75, 46, and 119, respectively. In order to verify the effectiveness of the HCNR, this chapter uses four attack strategies to attack the simulated network. The functional reliability is calculated as a function of different values of the strength coefficient of attack node. The results of the HCNR for the different attack strategies are shown in Fig. 5. The maximum cost B_{max} is set to 10% of the original network cost, and the maximum number of added edges n_L is set to 10% of the number of original network nodes, that is, $\varepsilon = 0.1$.

According to the simulation results, the repair algorithm is applicable to all the four attack strategies, and the functional reliability of the damaged network is improved by 0.16, 0.16, 0.16, and 0.12 on average, respectively. The repair effect is obvious. With the increase of f_N , as the total number of network nodes continues to decrease, the functional reliability of the network after repair also continues to decrease. As can be seen from the simulation results, among the four attack

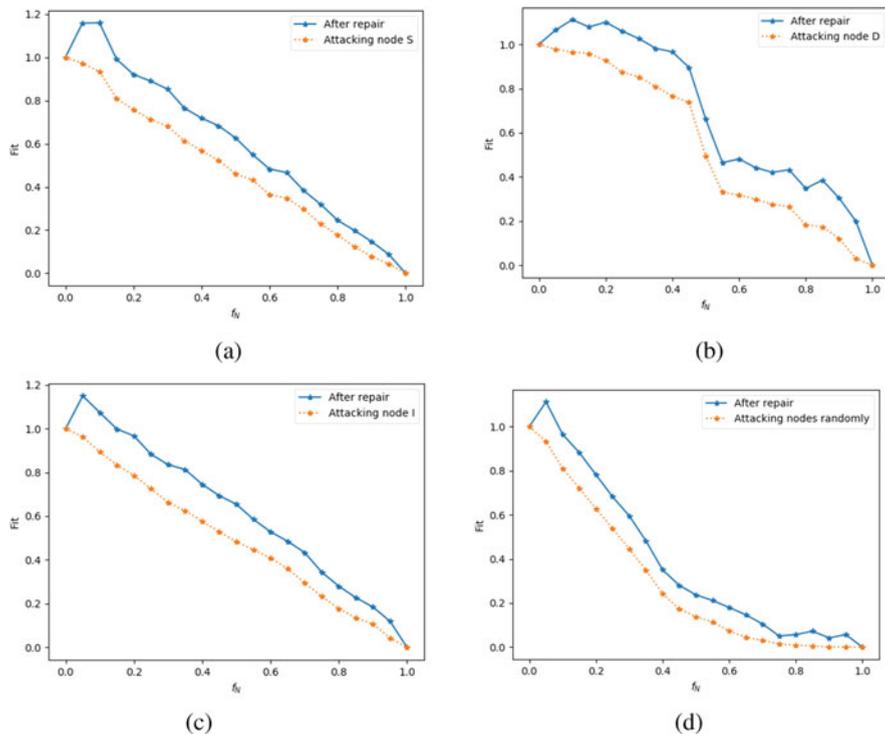


Fig. 5 The HCNR’s performance under four attack strategies. (a) Attacking sensor node. (b) Attacking decider node. (c) Attacking influential node. (d) Attacking node randomly

strategies, when f_N is higher than 0.7, 0.75, 0.7, and 0.35, respectively, the functional reliability of the network after repair is lower than 0.4. At that time, the network was seriously damaged and the repair was meaningless. In the analysis of all simulation results in this chapter, the data with functional reliability higher than 0.4 after repair are analyzed, that is, the meaningful repair results are analyzed.

2. Performance Comparison of Different Repair Algorithms

The HCNR is compared with the algorithm of random addition (RA), low degree first (LDF), and low betweenness first (LBF) [19]. Since the HCNR involves nodes with high degree, the high-degree first algorithm is also compared with the HCNR. The high-degree first algorithm (HDF) refers to the edge that meets the edge-adding condition between the nodes with the highest degree in the current network. Stop adding edges when the network connection cost or number of edges reaches the specified maximum.

The parameter setting is the same as that of simulation (1). As can be seen from Fig. 6, all the five algorithms can repair the damaged network and improve the functional reliability of the network. The results of HCNR are better than RA, LDF,

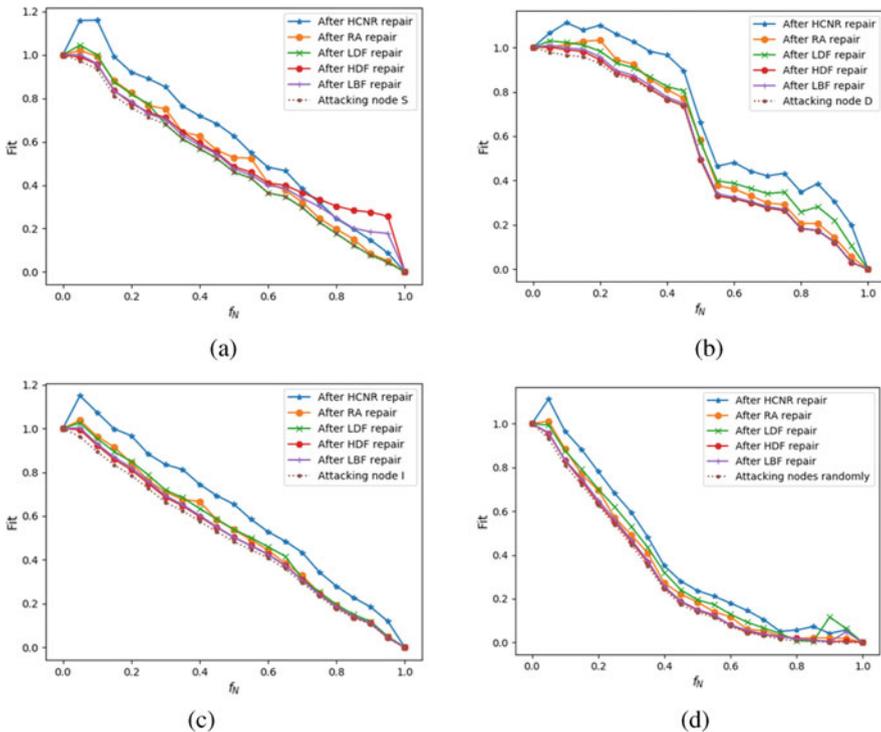


Fig. 6 Performance comparison of different repair algorithms under four attack strategies. (a) Attacking sensor node. (b) Attacking decider node. (c) Attacking influential node. (d) Attacking node randomly

HDF, and LBF. Under the four attack strategies, HCNR improves the functional reliability by 0.1 more than the other four repair algorithms.

3. Performance Comparison of Different Network Sizes

Figure 7 describes the average improvement rate obtained by 20 rounds of repair simulation according to four attack strategies when the number of network nodes is 140, 240, and 420, respectively. Among them, 20 rounds of data, mean variance less than 0.1, the simulation results are stable. Under the four attack strategies, the improvement rate of different network sizes is stable in the range of 13–32%, 13–40%, 13–35%, and 14–37%, respectively, with the increase of f_N .

Simulation results show that HCNR is suitable for different combat networks. As f_N increases, so does the improvement rate. The reason is that when f_N keeps increasing and approaching 1, the value of functional reliability of the damaged network decreases to a smaller value close to 0. At this point, due to the better repair effect, the rate of improvement is also relatively high. However, in the random attack

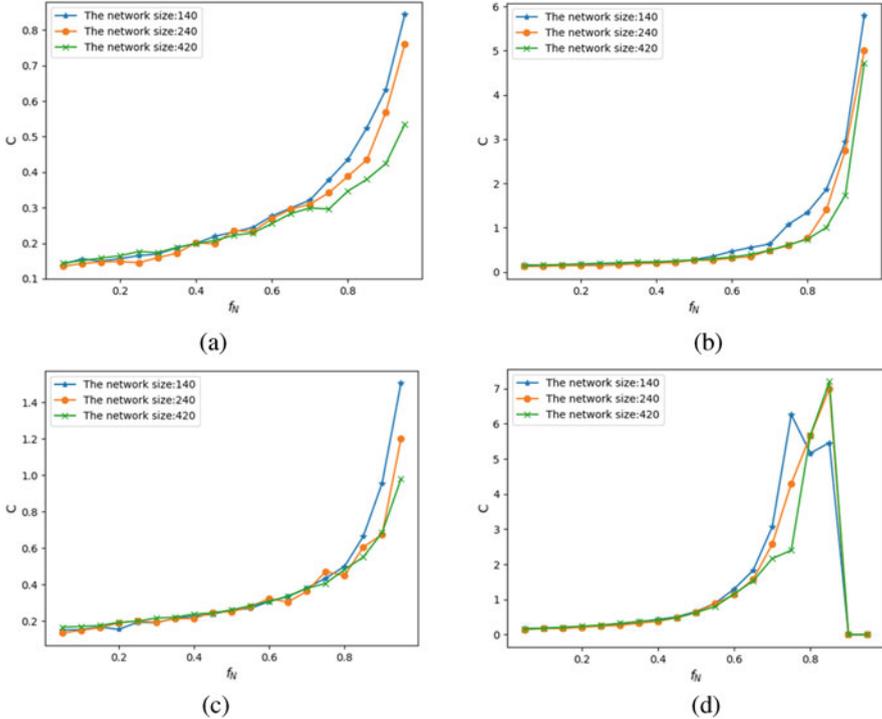


Fig. 7 The performance comparison of HCNR with different network sizes under four attack strategies. (a) Attacking sensor node. (b) Attacking decoder node. (c) Attacking influential node. (d) Attacking node randomly

strategy, after the node is seriously damaged, the network almost loses its combat capability, and the improvement rate is reduced to 0, as shown in Fig. 7d.

5 Conclusion

Based on the research of heterogeneous combat network, this chapter proposes a method to measure the functional reliability of combat network by combining network efficiency and network robust index. The HCNR is designed to improve the functional reliability of the damaged network. The best restoration scheme was obtained by using artificial colony algorithm. According to the nature of heterogeneous combat network, different attack strategies are used for simulation. The simulation results verify the effectiveness of the HCNR, which is applicable to different attack strategies, and prove that the proposed method is superior to other repair algorithms. It also shows that the HCNR is applicable to heterogeneous combat networks of different sizes.

Acknowledgments This work was supported by the Defence Advance Research Foundation of China under grant (no. 61400020109).

References

1. M. Zhang, Evaluation of Operational Capability of Land Force's Weapon Equipment System of Systems based on Function Chain. M.S. thesis, National University of Defense Technology, Hunan, China (2014)
2. Z. Qi, A study on the topology model of combat systems based on complex networks. *J. Ordn. Eng. Coll.* **25**, 66–69 (2013)
3. K. Chen, Y. Lu, Q. Liu, et al., A method to validate operational capability index model of heterogeneous combat networks based on characteristic topology analysis. *IEEE Access* **8**, 59760–59773 (2020)
4. M. Zhang, L. Ma, System-of-systems combat OODA loop robustness modeling and experiment. *J. Syst. Simul.* **29**, 1968–1975 (2017)
5. Q. Zhang, J. Li, D. Shen, Modeling and analyzing of battlefield information sharing effectiveness based on complex networks. *J. Syst. Simul.* **27**, 875–880 (2015)
6. Z. Miao, L. Ding, L. Zhao, et al., Network restoration based on topological information. *Comput. Eng.* **34**, 25–27 (2008)
7. B. Hu, F. Li, Repair strategies of scale-free networks under multifold attack strategies. *Syst. Eng. Electron.* **32**, 86–89 (2010)
8. X. Tian, Y. Zhu, K. Luo, et al., Adaptive reconstruction model for command and control system under information age based on complex network theory. *Syst. Eng. Electron.* **35**, 91–96 (2013)
9. L. Zheng, X. Li, J. Luo, et al., Research of repair strategies in complex network. *Electron. Des. Eng.* **22**, 140–142 (2014)
10. Y. Li, X. Wang, A repair strategy for scale-free network under the progressive intentional attack. *Electron. Des. Eng.* **25**, 181–184 (2017)
11. G. Chen, P. Sun, J. Zhang, et al., Repair strategy of military communication network. *J. Zhejiang Univ. (Eng. Sci.)* **53**, 1536–1545 (2019)
12. G. Peng, J. Wu, Optimal network topology for structural robustness based on natural connectivity. *Physica A Stat. Mech. Appl.* **443**, 212–220 (2015)
13. Y. Wang, S. Chen, C. Pan, et al., Measure of invulnerability for command and control network based on mission link. *Inf. Sci. Int. J.* **426**, 148–159 (2018)
14. Revay M, Liska M., OODA loop in command & control systems. 2017 Communication and Information Technologies (KIT), pp. 1–4 (2017)
15. X. Zhou, F. Zhang, W. Zhou, et al., Node efficiency is used to evaluate the robustness of complex network functions. *Acta Phys. Sin.* **61**, 7–13 (2012)
16. J. Li, Y. Tan, K. Yang, et al., Structural robustness of combat networks of weapon system-of-systems based on the operation loop. *Int. J. Syst. Sci.* **48**, 659–674 (2016)
17. J. Li, J. Jiang, K. Yang, Research on functional robustness of heterogeneous combat networks. *IEEE Syst. J.* **99**, 1–9 (2018)
18. B. Zhu, F. Zhu, H. Sun, et al., Discrete artificial bee Colony algorithm based on logic operation. *Acta Electron. Sin.* **43**, 2161–2166 (2015)
19. Z. Jiang, M. Liang, D. Guo, Enhancing network performance by edge addition. *Int. J. Mod. Phys. C* **22**, 1211–1226 (2011)

The Influence of Decorations and Word Appearances on the Relative Size Judgment in Viewers of Tag Clouds



Khaldoon Dhou, Robert Kosara, Mirsad Hadzikadic, and Mark Faust

1 Introduction

Visualization is concerned with studying, understanding, viewing, and communicating the visual representations of data. One must discern how eye and brain collaborate to process complex designs because misunderstanding of data can lead to serious consequences. For example, Elting et al. [6] found that the display format of data affects the accuracy of the decisions made by physicians. This makes it crucial to further investigate different visualizations to explore how they are perceived by viewers. In this chapter, we examine how changes in the presentation characteristics in tag cloud displays affect the perception of the relative importance of tag words in the display.

A tag cloud is a visual representation of the word content of a text (e.g., webpage, article, or book) which presents content words (i.e., tags) that vary in typeface size, or other features such as color or boldness, in relation to the frequency or importance of the content words in the underlying text. Tag clouds are becoming popular navigational tools for several reasons. They provide a summary of the relative importance based on the frequency of the appearance in keywords, and users can use tags to reach any desired section of text being referenced by a tag cloud.

K. Dhou (✉)
Texas A&M University Central Texas, Killeen, TX, USA
e-mail: kdhou@tamuct.edu

R. Kosara
Tableau Software, Seattle, WA, USA

M. Hadzikadic · M. Faust
UNC Charlotte, Charlotte, NC, USA

Typeface size is the most common and influential feature of tags used to communicate frequency or importance of words in the underlying text [4–6, 8, 9, 12]. It is also the nature of the tag cloud that viewers judge the words based on the relative size of other words in the same tag cloud. However, the extensive review of the existing literature revealed many unanswered questions on how viewers make relative judgments of the size of the words in a tag cloud. Not knowing how viewers perceive the relative typeface sizes in tag clouds is a critical issue, as biased or distorted perception of relative size will result in distorted perception of the frequency or importance of content words in the underlying text. Therefore, it is crucial for tag cloud designers to understand the relative typeface size judgments.

In this chapter, we focus on how users judge the relative size of words in a tag cloud in the presence of other design elements already studied in the literature: shape, decorations, and size of the target words in the tag cloud [2, 10, 18, 20]. More specifically, we investigate the following aspects:

- How do participants judge the relative size of words in tag clouds and how does that relate to relative size manipulation?
- How is this perceptual bias affected by the presence of a tag cloud context?
- How does the inclusion of different proportions of wide/narrow letters or letters with ascending or descending portions affect the perceptual bias in relative size judgment?
- How does the inclusion of decoration of the text box surrounding the tag affect the perceptual bias in relative size judgment?

2 Related Work

A tag cloud is a visualization that has drawn the attention of researchers from multiple fields. The typeface size is a primary visual characteristic in a tag cloud and is important in perception [2, 11, 19, 24]. Tag cloud designers use several other types of presentation characteristics that could interact with the size to influence relative size judgment such as the typeface size, weight, color, type, and decorations in the text boxes surrounding the tag [2, 11, 24]. Despite the growing literature on the importance of different features of tag cloud design, there are few studies reporting viewers' judgment of the relative word sizes in tag clouds. Dhou et al. [5] explored the influence of typeface size, weight, and word locations on judging the size of words in a tag cloud. In another study, Felix et al. [9] investigated different visual patterns to explore how they affect extracting the data from a set of keywords.

Studying the relative size judgments of words in a tag cloud has its grounds in psychophysics, the study of lawful relationships between physical aspects of objects and events (i.e., luminance, sound pressure, and light) and perception (i.e., brightness, loudness, and size). A main research finding from this literature is that ratio magnitude judgments, a ratio comparison, can be reliably made [3, 7]. This does not mean that such judgments are not systematically biased. For example,

Mates et al. [17] measured the ability of participants to compare the areas of squares and rectangles and found that rectangles were judged as bigger than squares, presumably due to overweighting of the longer dimension of the rectangle. Based on this finding, we predict that people might over-judge the size of words as they do for rectangles, but it is still unclear how this will translate into a judgment of the relative ratio of sizes of two compared words.

3 Experiment

The focus of the experiment is to attempt to understand how viewers perceive the relative size of the words given different characteristics and without including any effect of the semantics.

3.1 Participants

Participants ($N = 65$, 41 males and 24 females) who indicated normal or corrected 20/20 full color vision were recruited from the Amazon's Mechanical Turk [21], and they were compensated monetarily based on the accuracy of their responses during the relative size judgment task. All participants agreed to participate following an approved informed consent procedure.

3.2 Materials

Participants in the study were asked to judge the relative size of target word pairs made up of a target word and an anchor word for comparison. The design involved manipulation of four independent variables – tag cloud display type, appearance, decoration, and relative size:

- The presence or absence of a background tag cloud.
- Appearance of the letters in the target word pair was varied by varying the amount of letters with ascending and descending portions (e.g., h, c, p) and varying the amount of narrow and wide letters (e.g., w, i; see Fig. 1(b) for the three word pairs used in this experiment).
- Target word pairs differed in terms of decorations such as shading and boxes around the words (see Fig. 1(a)). For this manipulation, all words in a tag cloud including the pair of words to be judged have the same decoration.
- Relative typeface size of words of target word pairs was manipulated. The smaller anchor word was always presented in 12 pt. font, while the larger target word was presented in a varying font size (18, 24, 30, or 36 pt.).

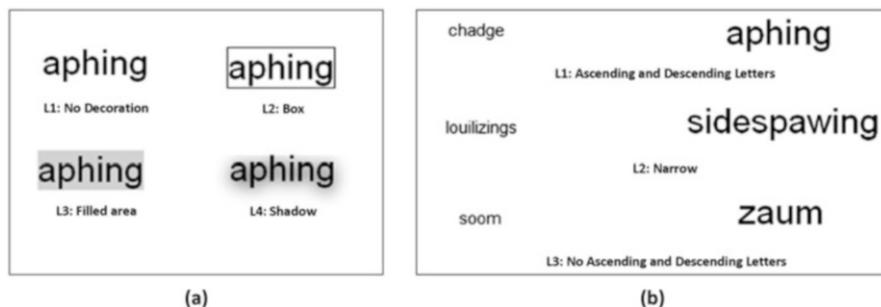


Fig. 1 (a) Decorations of words used in the experiment. (b) Appearance of words in a tag cloud

Forty-eight target pairs were constructed by taking the three pairs of words presented in Fig. 1(b) (originally from [22, 23]) and creating four decoration levels of each pair at four relative sizes of the larger word. Each target pair was presented twice (96 total displays), once with, and once without, a tag cloud background. Displays were displayed in a unique random order for each participant. To avoid the influence of semantics, we used words from Lorem Ipsum, which is a modified Latin text commonly used as filler in layout designs [13]. The criteria for choosing the target pair of words were the following:

- Both words have the same number of letters.
- Both words have the same number of ascending/descending letters.
- Both words have the same width when presented in same typeface size (except when proportion of wide and narrow letters is varied).

Tag cloud background words included 40 words from the same source [13]. These were randomly distributed and varied in size with a constant distribution. The target pair was always presented with the smaller anchor word in the left half of the display and the larger target word on the right. The pair of the words to be compared in the tag cloud present display type was marked by red dots (example on Fig. 2) as color is one of the ways to achieve emphasis via contrast on certain elements [16]. The screen size was always 800×600 pixels.

Each relative typeface size ratio judgment at each particular relative size was compared with the typeface size ratio squared at that particular relative size as this value is close to the area of the tag, which was defined as the minimal box around the word in a tag cloud [1].

3.3 Procedure

Participants judged each of the 96 displays, presented in a unique random order, with a brief break at the halfway point. They indicated their relative size judgment



Fig. 2 Screenshot of tag cloud present display type

for the target word pair in each display via a continuous scale, in tenths, ranging from 1 (same size) to 6 times larger. Displays remained visible until a response was completed.

4 Results

The relative size ratio judgments were submitted to a two display types (tag cloud present vs. absent) by three appearances (ascending and descending letters, wide vs. narrow, and no ascending and descending letters) by four decorations (no decoration, boxes, filled areas, and shadows) by four target word sizes (12 vs. 18, 12 vs. 24, 12 vs. 30, and 12 vs. 36) repeated-measures ANOVA.

4.1 Analysis of Full Design

The main effect of the display type was significant, $F(1, 64) = 6.305, p = 0.015, \eta^2 = 0.09$. This indicates that the mean size ratio judgment with the tag cloud present ($M = 3.34$) was significantly larger than the mean size ratio judgment with

the tag cloud absent display type ($M = 3.21$). There was also a significant interaction effect between the display type, appearance, and target word size, $F(6, 384) = 2.239$, $p = 0.039$, $\eta^2 = 0.034$. The significant interaction display type, appearance, and typeface had a very small effect size and did not involve an appreciable qualitative change in the pattern of means between displays with and without a background tag cloud present.

4.2 Tag Clouds Present Display Type

A $3 \times 4 \times 4$ repeated-measures ANOVA was conducted including only the displays with tag cloud backgrounds. The main effect of the typeface size of the target word was significant, $F(3, 192) = 285.501$, $p < 0.001$, $\eta^2 = 0.817$. Additionally, the main effect of the appearance was significant, $F(2, 128) = 45.674$, $p < 0.001$, $\eta^2 = 0.416$, with the mean ratio judgment being significantly larger with the inclusion of wide letters ($M = 3.572$) than with inclusion of ascender/descender letters ($M = 3.233$) or with inclusion of neither ($M = 3.211$, post hoc comparisons, $p < 0.05$). While there was a significant interaction between appearance, decoration, and size, $F(18, 1152) = 1.766$, $p = 0.025$, $\eta^2 = 0.027$, this higher order interaction did not influence the interpretation of the component two-way interaction (presented in Fig. 3). The interaction between appearance and size was significant, $F(6, 384) = 2.286$, $p = 0.035$, $\eta^2 = 0.034$, as was the interaction between decoration and size was likewise significant, $F(9, 576) = 2.197$, $p = 0.021$, $\eta^2 = 0.033$. The interaction between appearance, decoration, and size was also significant, $F(18, 1152) = 1.766$, $p = 0.025$, $\eta^2 = 0.027$.

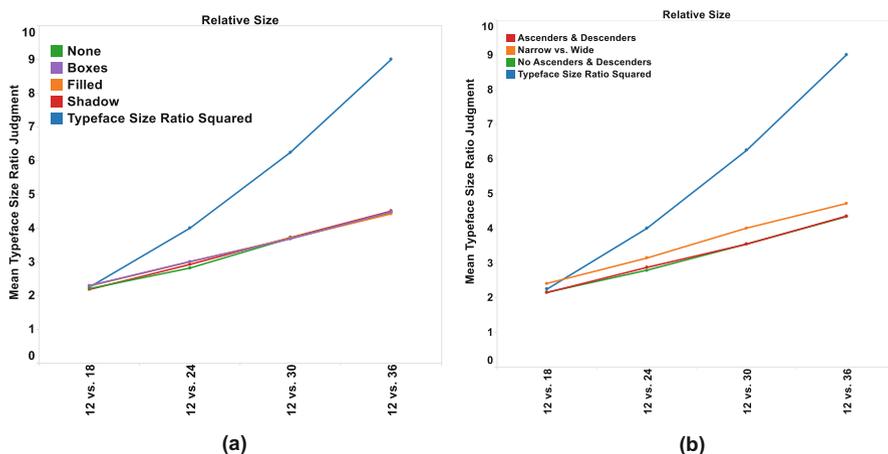


Fig. 3 (a) Interaction between decoration and size. (b) Interaction between appearance and size for the tag cloud present display type

5 Discussion

As can be seen in Fig. 3, there was a large bias toward the underestimation of the relative size ratio of target word pair that increased with size. This large underestimation bias was modestly affected by the addition of wide letters to the target word, supporting the idea that the length dimension of the (roughly) rectangular words was a stronger influence on the perceived size of the word than was the height dimension. It is worth noting that the addition of ascender/descender letters to the target word did not have a significant influence on the underestimation bias in this study. Also of note is the fact that adding decorations (e.g., shading) to more explicitly define the rectangular text box surrounding each word, had only a minor influence on the underestimation bias, and only for a range of smaller relative sizes for the target word pair. Finally, there was a small main influence of the presentation of a tag cloud background to reduce the underestimation bias somewhat.

The substantial underestimation bias may have been partly due to an anchoring effect [14]. That is, use of a single typeface font for the anchor words, that was at the smallest size used for the tag cloud as a whole, may have provided an anchor point that influenced the range (at the lowest relative size for the target pair of 2.0) point where relative size ratio judgments were approximately veridical. It remains to be seen if increasing the size of the anchor word will influence the range of veridical relative size judgment.

The experiment revealed some findings on the appearance of words in tag clouds. Viewers seem to not be influenced with ascending and descending letters, suggesting that tag cloud designers should not put a lot of emphasis on controlling the existence of ascending and descending letters in tags as they do not seem to add to the relative size judgment bias documented in the present study. By contrast, this study clearly showed that the size ratio judgment was greatly influenced by varying the length between the two words in the target pair, which is an indication that the length factor needs to receive more attention in the creation of tag clouds.

Although mean typeface size ratio judgments between different decorations seem to be close to each other, empirical results showed a minor influence of boxes and filled areas on the typeface ratio size judgment when a small target word was used. This correlates with what has actually been applied in journalism where decorations around the words were used to emphasize small typeface size [15]. Based on the empirical results in this chapter, decorations may reduce the general underestimation bias for target words with small font sizes.

6 Conclusion

The results demonstrated approximately veridical relative size ratio judgment for a relative size of two (target word text box twice the size of the comparison anchor word), and a rapidly increasing large underestimation of relative size ratio as the

size of the target word increased. This underestimation bias was only modestly affected by increasing the number of wide letters (e.g., w, m) in the target word, in comparison to the comparison anchor word, and was mostly unaffected by other variations in appearance and decoration of the text box attempted. Future research should examine the variation of the size of the anchor word and the location of the target word pair in the tag cloud to verify these findings. The results have clear applied implications for design of tag clouds and have implications for visualization techniques that depend on perception of the relative size of words to convey the important interpretive information. Future research includes investigating different characteristics of tag clouds and how they can influence the judgments of viewers. Examples of these characteristics include color, location, and semantics.

References

1. S. Bateman, (Oct 2013), personal correspondence by email
2. S. Bateman, C. Gutwin, M. Nacenta, Seeing things in the clouds: The effect of visual features on tag cloud selections. In: Proceedings of the nineteenth ACM conference on Hypertext and hypermedia. HT'08, ACM, New York, NY, USA (2008), pp. 193–202
3. W.S. Cleveland, C.S. Harris, R. McGill, Human factors and behavioral science: Experiments on quantitative judgments of graphs and maps. *Bell Syst. Tech. J.* **62**(6), 1659–1674 (1983)
4. K. Dhou, Toward a Better Understanding of Viewers' Perceptions of Tag Clouds: Relative Size Judgment. Ph.D. thesis, USA (2013)
5. K. Dhou, M. Hadzikadic, M. Faust, Typeface size and weight and word location influence on relative size judgments in tag clouds. *Journal of Visual Languages & Computing* **44**, 97–105 (2018) <http://www.sciencedirect.com/science/article/pii/S1045926X16300210>
6. K.K. Dhou, R. Kosara, M. Hadzikadic, M. Faust Size judgment and comparison in tag clouds. *IEEE Visualization Poster Proceedings* (2013)
7. G. Ekman, K. Junge, Psychophysical relations in visual perception of length, area and volume. *Scand. J. Psychol.* **2**, 1–10 (1961)
8. J. Emanuel, The Millennials: Assessing the next generation of academic librarians. Ph.D. thesis, University of Missouri-Columbia (2012)
9. C. Felix, S. Franconeri, E. Bertini, Taking word clouds apart: An empirical investigation of the design space for keyword summaries. *IEEE Trans. Vis. Comput. Graph.* **24**(1), 657–666 (2018)
10. M. Furini, On introducing timed tag-clouds in video lectures indexing. *Multimed. Tools Appl.* **77**, 967–984. Springer Nature (2017)
11. M.J. Halvey, M.T. Keane, An assessment of tag presentation techniques. In: *Proceedings of the 16th International Conference on World Wide Web*. ACM (2007), pp. 1313–1314
12. K. Hoyt, Changing the climate while reproducing power?: Examining the social construction of renewable frames in mass print news media. Ph.D. thesis, University of Colorado (2012)
13. L. Ipsum, Lorem Ipsum. (May 2011), <http://www.lipsum.com/>
14. K.E. Jacobowitz, D. Kahneman, Measures of anchoring in estimation tasks. *Personal. Soc. Psychol. Bull.* **21**, 1161–1166 (1995)
15. Keepr, Keepr. Online (October 2013), tag cloud retrieved from www.keepr.com on 21 Oct 2013
16. D. Lauer, S. Pentak, *Design basics* (Cengage Learning, Boston, 2007)
17. J. Mates, V. Di Maio, P. Lansky, A model of the perception of area. *Spat. Vis.* **6**(2), 101–116 (1992)

18. D.S. Pathak et al., Visual exploration based comparative analysis of tag cloud layout. *Networking and Communication Engineering* **8**(2), 43–46 (2016)
19. A.W. Rivadeneira, D.M. Gruen, M.J. Muller, D.R. Millen, Getting our head in the clouds: toward evaluation studies of tagclouds. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. CHI'07, ACM, New York, NY, USA (2007), pp. 995–998
20. M.E. Sereno, K.E. Robles, A. Kikumoto, A.J. Bies, The effects of three-dimensional context on shape perception. *Psychol. Sci.* **31**(4), 381–396 (2020). <https://doi.org/10.1177/0956797620901749>, pMID: 32163718
21. A.M. Turk, Amazon Mechanical Turk. <http://mturk.com/> (November 2011)
22. N. Words, Non-sense words. (May 2011), <http://www.soybomb.com/tricks/words/>
23. N. Words, Nonsense words for use in training phonemic awareness. (May 2011), <http://www.speech-language-development.com/nonsense-words.html>
24. Q. Zhang, W. Qu, L. Wang, How font size and tag location influence Chinese perception of tag cloud? in *Engineering Psychology and Cognitive Ergonomics*, (Springer, 2011), pp. 273–282

Automation of an Off-Grid Vertical Farming System to Optimize Power Consumption



Otto Randolph and Bahram Asiabanpour

1 Introduction

As the world's population increases, resources must be stretched and used more efficiently than ever to provide the same living standards people are used to. Even basic resources, such as land and water, are strained. If vertical farming techniques were adopted, vast areas of land would be freed up for other uses. Water and fertilizer use would drop dramatically [2]. The problem is that vertical farms harm the environment through their intensive energy use. Vertical farms replace sunlight and rain with electric lights and water pumps. The more plants in the system, the more these demands increase.

The best solution to this problem involves creating an automated, off-grid vertical farming system powered by renewable energy, such as wind or solar. Using a renewable energy source reduces the negative consequences of the high energy needs of a vertical farm. Automating the system allows it to adapt to changing weather conditions and their effects on power production. This improves the efficiency of the whole system and reduces the need for expensive and environmentally harmful batteries. This can be done because plants adapt and thrive in a wide variety of growing conditions.

While there have been many attempts to automate vertical farming systems, most occur in a laboratory setting, and none combine full system automation with an off-grid power source. One group of researchers automated the watering and fertilization schedule of their plants [1]. Others automated the watering schedule of a full-sized system [2]. Some went so far as to patent their automatic watering and fertilization system. However, none of these were off-grid systems.

O. Randolph (✉) · B. Asiabanpour
Texas State University, San Marcos, TX, USA
e-mail: ocr4@txstate.edu; ba13@txstate.edu

While there are some off-grid farming systems, most are limited in scope. Some are primarily used to monitor plants growing in remote locations [5]. Other network off-grid systems create a more stable micro-grid, but they still require a grid connection to ensure power stability [4]. The closest system found was an off-grid aquaponics system that used solar energy to power pumps that circulated water from fish tanks to plants [3]. While close to being a vertical farm, this system was not an actual vertical farm because it relied on sunlight to grow the plants.

2 Methods and Materials

The system needed to be automated because a system-wide power outage could be catastrophic. Short-term lack of water causes plants to wilt while heat buildup from lack of air-conditioning can directly kill plants or cause them to go to seed. Automating the system could stop a system-wide power outage by turning off nonessential services, preserving a reserve amount of battery power to keep things running smoothly.

The primary sources of energy use are the lights, air-conditioning, fans, and pumps. Lights use the most power and are least critical for the short-term health of plants. Air-conditioning is vital to the plants during hot days but is not needed during the winter months. Running the AC at full power uses the same amount of energy as one half of a rack of lights. Fans are nonessential but serve as a supplementary cooling system if the outside air is cooler, using a negligible amount of power. Water pumps are essential, but use a negligible amount of power. Therefore, an ideal system would turn off lights first, air-conditioning next if it were in use, and save fans and pumps for last. In the event of an unavoidable power outage, the system should have the ability to be quickly restarted if not automatically restarted.

The proposed system had to meet several requirements to automate it successfully. First, it was integrated with the existing fans, pumps, AC system, and lights. Second, it had to interpret data and make decisions from it. Third, the automation had to be compatible with a range of external sensors and support a Modbus connection to the solar panel system. Finally, it needed to be inexpensive, reliable, and have a large programming community in case problems arise.

Experimentation on the system showed that power reliability in the system had the most significant effect on plant health. If the power supply is unreliable, the system experiences power outages. During a power outage, all environmental controls are off. In the short term, the system could overheat, stunting the plants and impacting produce quality. In the long term, the plants could die from lack of water. Therefore, the primary method used to compare the automated and timed systems was power reliability.

3 Results

Two strategies were used to compare the effects of automating the system versus having a fixed schedule system. Strategy one simulated a system run by timers and compared the power stability to the automated system. Racks were run for 8 hours per day, and the number of racks each month was chosen based on the results of an idealized simulated system.

The long-term decision on how many plants to grow each month was determined by an Arena simulation model of the system. In it, historical power production data were fed into a model of the automated system. As power was available, portions of the system were turned on or off. The model had three limits. First, in low power conditions, it had to reduce power usage to ensure the minimum growing conditions were met and prevent a power outage. Second, the limiting factor for power use was the inverter. Power use could not exceed the inverter limits, but excess energy above inverter limits could be stored in the batteries until they were full. Third, if excess power were available, subsystems would be turned on until the inverter limit was reached. These ideal results were then used to determine the number of plants that could be grown each season. Because the number of racks in operation determines the number of plants grown, the results of the simulation were found in racks of plants grown, Fig. 1. These numbers were used as baseline statistics to determine the number of racks of plants to grow each season.

The number of racks of plants grown each month was chosen based on the simulation. The automated system adapted to changing power availability each day

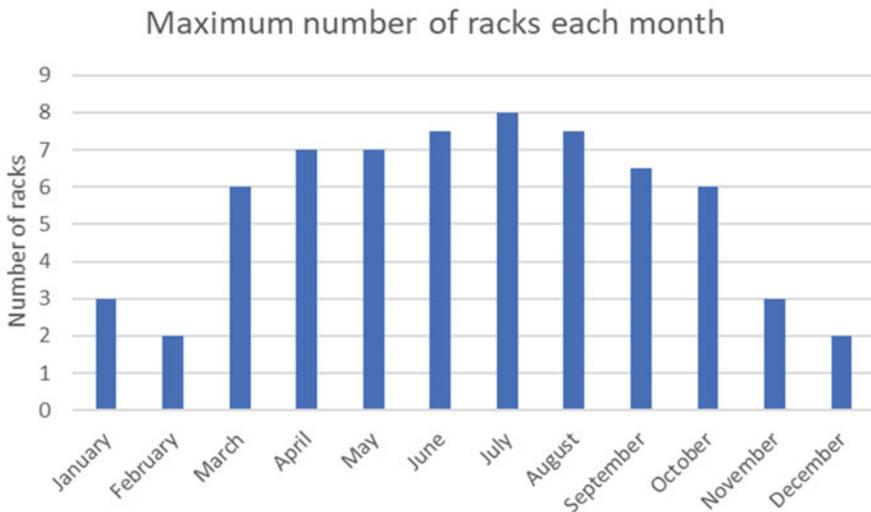


Fig. 1 The maximum number of racks of plants that can be grown each month, in half-rack intervals

and never experienced a power outage. The base system was run on a fixed, timed schedule. Instead of running the lights and air conditioner as power was available, the lights and AC were run for 8 hours per day at fixed times. Lighting schedules were staggered to better match when the power was generated and to reduce the likelihood of using up the battery reserves during non-peak energy production hours.

The data from four different months were run through the modified program to see how the timed system would react. The 4 months of run were January, April, July, and October, representing 1 month from each season. In January, two racks were run. In April and October, six racks were run. In July, eight racks were run. The results showed that while the timed system did not exhibit 100% power stability for all months, it did maintain stability most of the time. Figure 2 shows the full results. While this was promising, the problems were more significant. When a power outage occurred in the physical system, the timers stopped. When power was returned, the timers were now out of sync with energy production, creating even more power outages. This did not happen in the simulation. Instead, the timers were always synced with the ideal schedule.

Strategy two automated the system and monitored the power stability for 2 months. Two strategies were considered for system automation, a network of Raspberry Pi microcomputers or an industrial-grade programmable logic controller (PLC). The PLC was a professional system and ideal for the final system. The Raspberry Pi system was chosen because of its simplicity, low cost, extensive support database, and ability to integrate into existing hardware.

The Raspberry Pi network was divided into two parts, the hardware and the control algorithm. The hardware consisted of a central hub Pi that controlled servant Pis through a Wi-Fi connection, Fig. 3. The hub Pi ran the control algorithm, collected temperature data from the shipping container Pis, and collected the state of charge (SOC) and net power flow of the batteries from the solar panel setup via

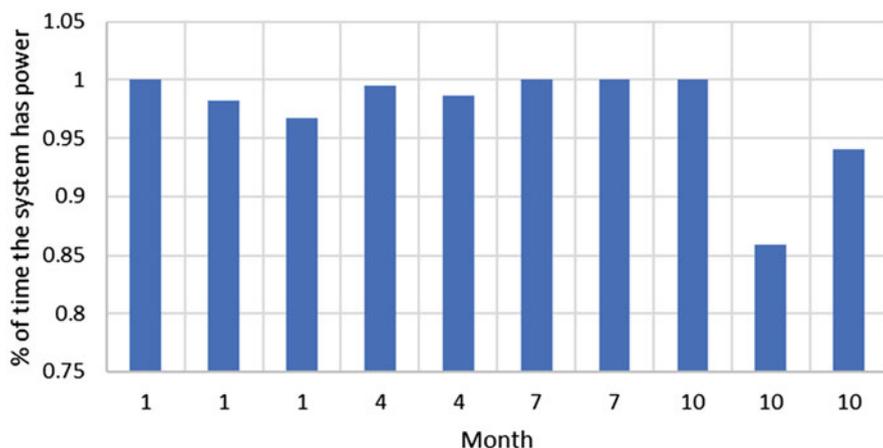


Fig. 2 Power stability each season

a Modbus connection. These data were fed into the control algorithm to determine whether the AC should be turned on, as well as the number of lights that could be turned on. Once the appropriate response was determined, the algorithm sent out the instructions to the servant Pis. Each servant Pi controlled one to two power outlet relays via a hardline connection. In the event of a power disruption, the servant Pis were programmed to keep the outlets off until the connection was restored, preventing further power failures. The only Pi not networked to the system was the fan control Pi, Fig. 4. This Pi monitored the internal temperature of the container and the external ambient air temperature. If the internal temperature was higher than the external temperature and the internal temperature was greater than 65 °F, the fan turned on to cool the system. This was done to reduce complexity, and because the external fan used a negligible amount of power.

The control algorithm also had two parts. Part one determined the number of devices to run, Fig. 5. Due to inverter limits, a total of 12 devices could be run at one time. One device was classified as a single air-conditioning unit or half of a rack of plants. If excess power was available (limited energy flow out of the batteries of less than 2 amperes), the system turned on an additional device. If excess power was not available (more than 2 amperes of power flowing from the batteries), the algorithm

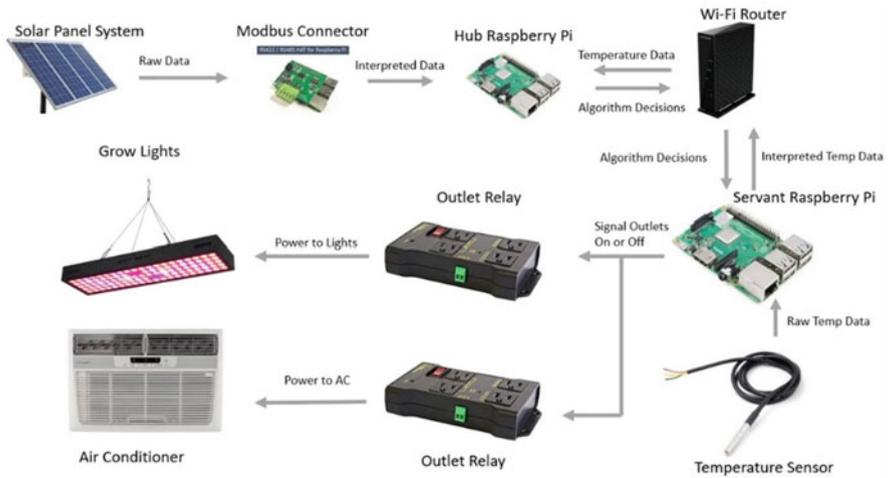


Fig. 3 How the main system is connected

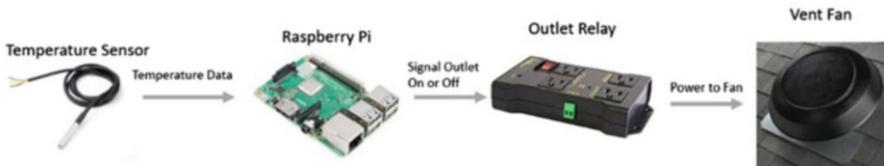


Fig. 4 Fan control system

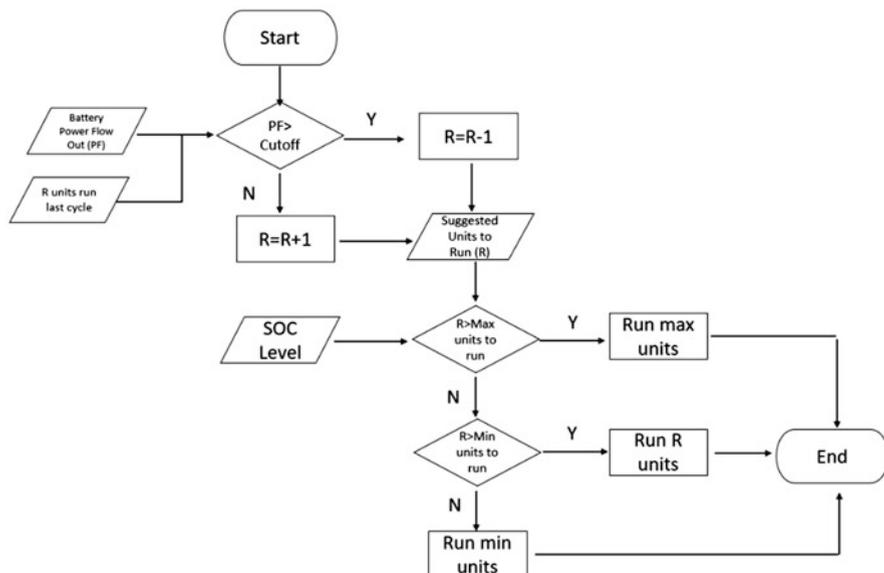


Fig. 5 Algorithm to determine the number of units to run

turned off one device. This initial response was then modified based on which of the five levels the battery state of charge (SOC) was in. If the battery state of charge was below 10%, the system turned off all nonessential subsystems, leaving only the fan and pumps running, as necessary. Between 10% and 30% power, the system ran the air conditioner as needed and ran up to half capacity if the batteries were actively being charged. Between 30% and 50% battery SOC, the system ran up to half-power if excess power was available. Between 50% and 90% SOC, the system ran at least half-power, going up to full power if excess power was available. Because charging the batteries does not count toward inverter capacity, running down the batteries each day to 50% SOC increased the available energy in the system up to 5 kWh on sunny days. At 90%+ SOC, the system slows down charging the batteries, and excess power is wasted. To prevent power wastage, the system runs at full power irrespective of whether energy is flowing into or out of the battery pack.

Once the system determined the number of devices to run, it went down the list of available devices, Fig. 6. The air conditioner only ran when the internal temperature was above a certain threshold. If the temperature was below this threshold, the subsystem was not considered. Next, each half rack of lights was considered. The system tracks how long each half rack has run that day. If it has run longer than the user-specified time, it is ignored. If it has not, it is powered on. This process was repeated until every available device powered on, or all available power was allocated. Once this was done, the algorithm sent out the decisions and waited 60 seconds before repeating the cycle.

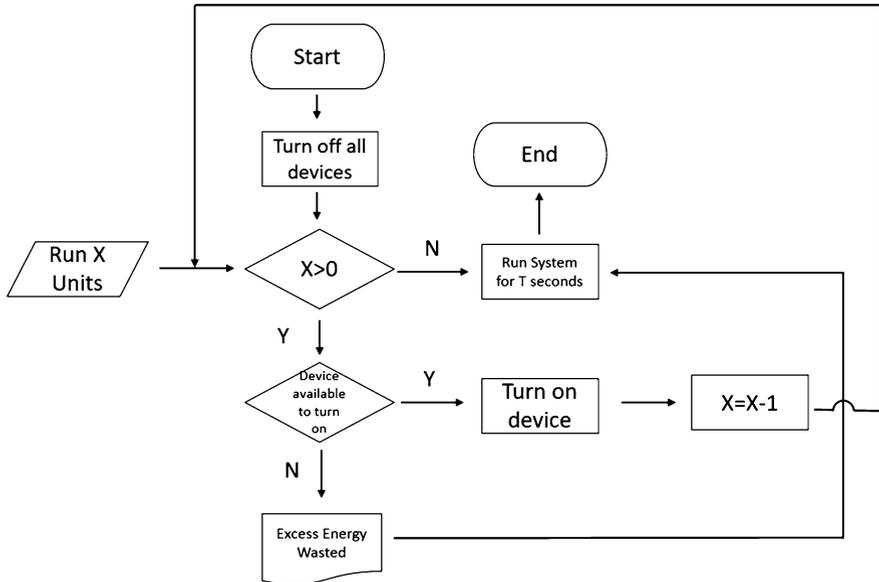


Fig. 6 How power is distributed in the system

The Raspberry Pi network was installed over 1 month, and then run for 2 months. Over this time frame, the system only lost power twice, once when the system was partially implemented and once when the system was fully operational. The power outages were discovered several hours after the fact, but in both cases, power was restored within minutes, meaning any timers in the system were not out of sync. Furthermore, these power outages occurred after the worst possible weather conditions, days of cloudy, rainy weather where energy generation was minimal. If the system had not been automated, it would have been out of power for hours or days and required a full reset of all the timers.

4 Conclusion

The world needs to be more efficient with natural resources if it wants to maintain the same standard of living for future generations. Off-grid, renewable energy powered, vertical farms could be one of the solutions, but the lack of power stability within the physical system proved fatal for the plants within them. The main way to kill plants was to remove their access to water, as found during a power failure. The simulation showed that an automated system would never fully deplete the battery reserve. When a timed system was modeled, at best the system would have perfect power stability, but a power failure would occur most months, Fig. 2. When

a physical system was implemented, this was shown to be true. The fully automated system adapted to changing weather conditions without suffering a major power outage. If the fixed, timed system had still been installed, it would have suffered a major power outage and would have lost power for several hours until it was reset. Therefore, an automated system can adapt to changing weather conditions without causing a power failure or killing the plants, as shown in the simulated and physical systems. This makes off-grid vertical farms a practical solution for commercial farming.

Acknowledgments This work has been completed with funding from the US Department of Agriculture (grant number 2016-38422-25540). The authors would like to thank the USDA, Freeman Center, and Ingram School of Engineering at Texas State University for providing funding and access to infrastructure and laboratories. Sponsors are not responsible for the content and accuracy of this article.

References

1. P. Belhekar, A. Thakare, P. Budhe, U. Shinde, V. Waghmode, *Automated System for Farming with Hydroponic Style*. Paper presented at the Fourth International Conference on Computing Communication Control and Automation (ICCUBE), Pune, India. (2018). Conference retrieved from <http://libproxy.txstate.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edsee&AN=edsee.8697884&site=eds-live&scope=site>
2. K.-Y. Choi, E.-Y. Choi, I.S. Kim, Y.-B. Lee, Improving water and fertilizer use efficiency during the production of strawberry in coir substrate hydroponics using a FDR sensor-automated irrigation system. *Horticult. Environ. Biotechnol.* **57**, 431–439 (2016)
3. F. Ismail, J. Gryzagoridis, *Sustainable development using renewable energy to boost aquaponics food production in needy communities*. Paper presented at the Proceedings of the Conference on the Industrial and Commercial Use of Energy, ICUE, Cape Peninsula University of Technology, Bellville, South Africa (2016, 2016/10/21)
4. C.G. Labrador, A.C.L. Ong, R.G. Baldovino, I.C. Valenzuela, A. B. Culaba, E.P. Dadios. *Optimization of power generation and distribution for vertical farming with wireless sensor network*. (2019, 2019/03/12)
5. J.E.M. Salih, A.H. Adom, A.Y.M. Shaakaf, Solar powered automated fertigation control system for Cucumis Melo L. cultivation in green house. *APCBEE Procedia* **4**, 79–87 (2012). <https://doi.org/10.1016/j.apcbee.2012.11.014>

Workflow for Investigating Thermodynamic, Structural, and Energy Properties of Condensed Polymer Systems



James Andrews and Estela Blaisten-Barojas

1 Introduction

Polymeric nanoparticles with hollow core such as those formed from poly(lactic-co-glycolic acid) (PLGA) constitute a vast class of soft matter nanostructures of exceptional technological significance for drug delivery and tissue engineering applications [1, 2]. PLGA is a biocompatible and FDA-approved biodegradable copolymer that has been used for the controlled delivery of antibiotics and macromolecules such as DNA and RNA. The benefits of such a delivery system are multiple, including applications in vaccines. Drug dosages and speed of delivery have been studied empirically by tuning the synthesis for specific physical properties of the condensed polymer phases. However, there is no systematic study at the atomic level of PLGA glassy solid, rubbery/leather region, and liquid phases.

In this work we describe a computational protocol for the scientific analysis of PLGA from the perspective of all-atom simulations of the PLGA polymer system. The workflow concatenates a collection of molecular dynamics packages, computational chemistry software, and numerous custom codes and database tools developed in a high-performance computing environment specifically for investigating the thermodynamic, structural, and energy properties of PLGA. This workflow is applied for a system of PLGA(50:50) with polymer chains containing 222 monomers.

J. Andrews · E. Blaisten-Barojas (✉)

Center for Simulation and Modeling and Department of Computational and Data Sciences,
George Mason University, Fairfax, VA, USA

e-mail: jandre17@masonlive.gmu.edu; blaisten@gmu.edu

<http://cmasc.gmu.edu>; <https://cds.gmu.edu>

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_75

1023

2 Methodology Protocol and Results

PLGA is a copolymer composed of two kinds of monomers, lactic acid (L) and glycolic acid (G), of which lactic acid has a pair of stereo-isomers, the L- and D-lactic acid. Initial polymer chains were constructed using a custom computer code able to create arbitrary length polymer chains with an arbitrary sequence of monomer types. The unoptimized chemical structures of L-L, D-L, and G monomers were from PubChem. A polymeric matrix of PLGA(50:50) was built containing a certain number of chains, each chain formed by an equal proportion of L and G monomers randomly sequenced along the chain. In these polymer chains, half of the L monomers were of type L-L and the other half of type D-L. For this study ten chains of 24 monomers each were created as described, completing the workflow step identified as *monomer sequence* in Fig. 1. The ten 24-mer chain models were energetically optimized at the density functional theory level using the package Gaussian09 [3]. These optimized structures and corresponding atomic charges are ported to antechamber (AmberTools package) [4] to generate parameters for the GAFF force field. Our custom script allows for selection between two schema of atomic charges: the restrained electrostatic potential (RESP) or the bond charge correction (BCC) method (details in [5]). This process completes the other two workflow steps identified as *atomic charge refinement* and *force field* in Fig. 1.

Longer chains of 222-mers were coalesced by stitching together combinations of the 22 central monomers of the different 24-mer chain models using leap (AmberTools package) [4]. A total of twelve polymer chains of molecular weight 14,459 u were generated. These stitched chains were relaxed and allowed to acquire random coil conformations in vacuum. Currently the four described steps of the workflow are implemented in an embarrassingly parallel workload manner. Thereafter, these 222-mer chains populated a computational box for modeling a PLGA(50:50) system with 20,016 atoms, completing the workflow step *structural optimization* in Fig. 1. The latter constituted the actual initial system configuration. Thereafter, in step *NPT annealing* of Fig. 1, the GROMACS package [6] was used for equilibrating the system with molecular dynamics (MD) within the constant number of particles N, constant pressure P, and constant temperature T (MD-NPT) approach at a high temperature of $T = 500$ K and $P = 101.325$ kPa. The MD parameters were periodic boundary conditions, time step of 1 fs, cutoff radius of 1.4 nm, Berendsen temperature and pressure couplings, and PME (particle mesh Ewald) for long-range electrostatics, and the system was set to run for 20 ns. Aided with a battery of scripts for slurm (job scheduler), 30 MD-NPT sequential runs were concatenated for annealing the system in a descending ladder-wise fashion from 500 K to 200 K. Each simulated temperature followed from the ending configuration of the previous temperature simulation. The fundamental outcome of the stepwise temperature annealing was the generation of the caloric curve of enthalpy as a function of temperature (Fig. 2a), its fluctuations (Fig. 2b), its Legendre transform (Fig. 2c), and the system volume as a function of temperature (Fig. 2d), which constituted the workflow task *isobaric analysis* of Fig. 1. A custom optimization

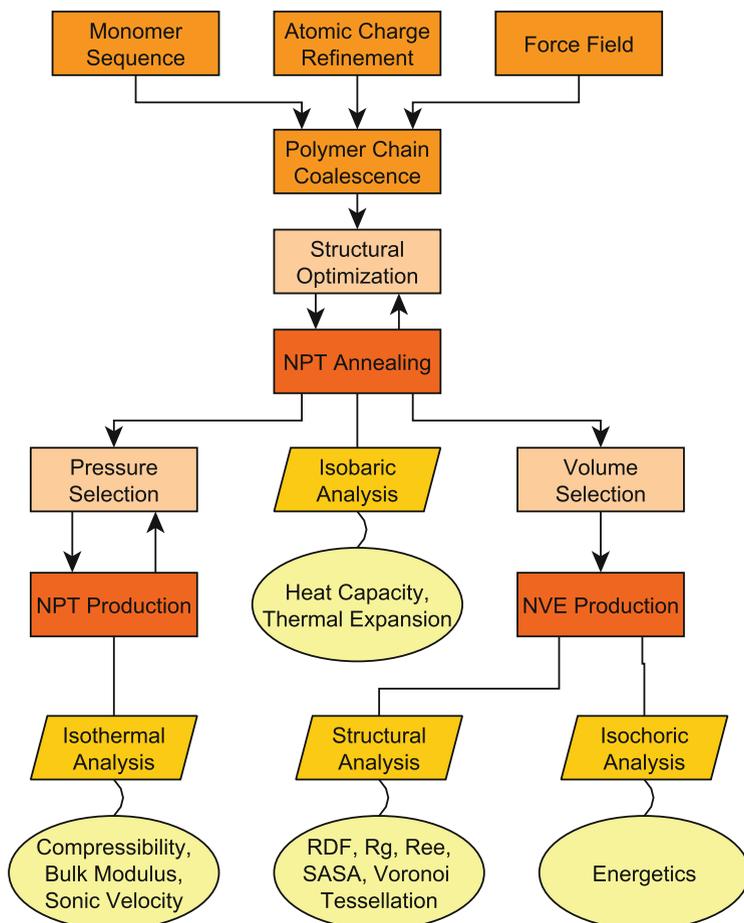


Fig. 1 Workflow of tasks leading to the all-atom inspection of the PLGA(50:50) polymer in its condensed phases

routine for identifying the inflection point between linear regions of the enthalpy, enthalpy variance, and volume led to a predicted glass transition temperature (T_g) and its bracketing temperature region between $T_{g\alpha}$ and $T_{g\beta}$ along which the polymer matrix transitioned from a viscosity-dominated system to a mechanically elastic material. The final step of the workflow central block was *specific heat and thermal expansion*, which included the calculation of both properties from the change of enthalpy and of volume with respect to temperature, respectively.

From here on, the workflow tasks split into two branches emanating from the step *NPT annealing* of Fig. 1. For the right-hand-side branch, the strategy was to use a round-robin scheduler in which as soon as one of the MD-NPT ladder-descending temperature runs finished, a new MD-NVE simulation at the

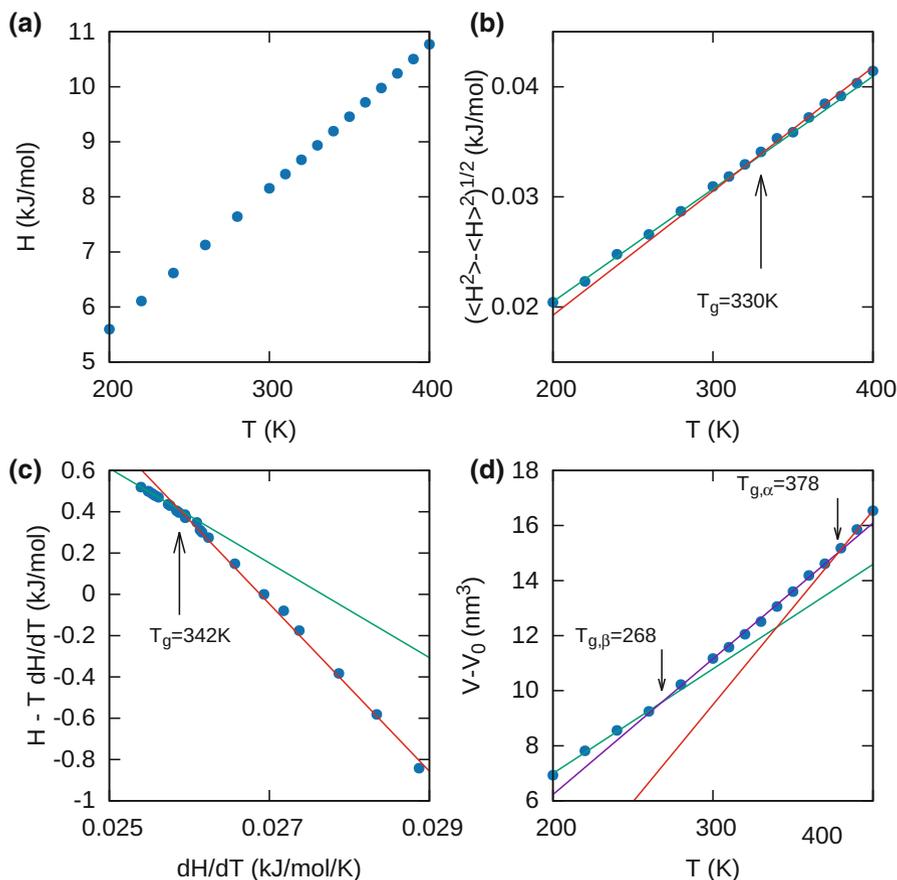


Fig. 2 PLGA enthalpy H as function of temperature (a); the enthalpy standard deviation as a function of temperature (b); Legendre transform of $H(T)$ (c); and system volume V related to the volume at low temperature V_0 (d). The glass transition temperature T_g is depicted with arrows. These functions were obtained using the BCC atomic charges in the GAFF force field

equilibrated volume was initiated and run for 20 ns. The latter involved steps *volume selection* and *NVE production* depicted in Fig. 1. Next, two parallel tasks in the workflow, (1) *isochoric analysis* and (2) *structural analysis*, led (1) to energy-derived properties obtained with custom-built codes (cohesive energy, cohesive energy density, Hildebrand parameter) and (2) to radial distribution functions (rdf) and solvent-accessible surface area (SASA) calculated with GROMACS-accessible tools. In addition, custom analysis codes and scripts were written for polymer chain radius of gyration (Fig. 3), moments of inertia, configuration tensor, and chain volumes, among other chain-based properties. Meanwhile, the left-hand-side branch of the workflow relates to a process designed for obtaining the polymer system behavior at a select set of pressures with the final goal of determining the isothermal

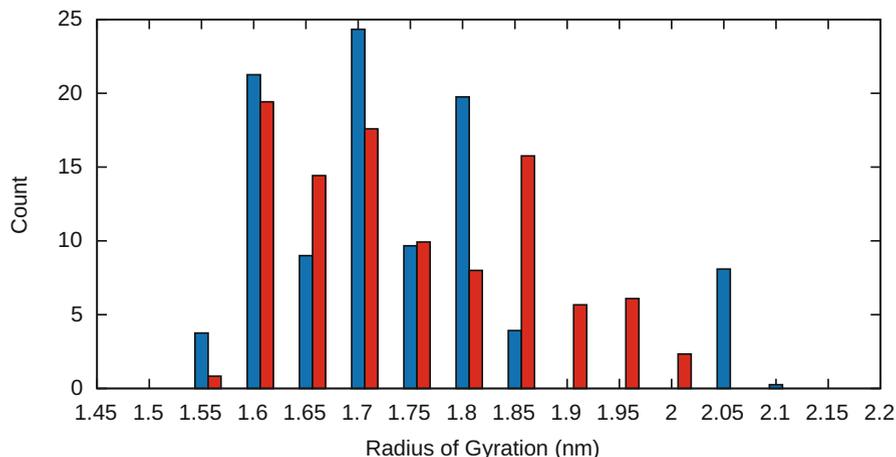


Fig. 3 Distribution of the radius of gyration collected from MD-NVE runs of 20 ns at $T = 300$ K (blue) and $T = 500$ K (red). The histogram collects R_g values for the 12 chains of length 222-mers along a 20 ns MD-NVE run

compressibility of PLGA(50:50). This task entailed several MD-NPT runs at six pressures in the range 500–50000 kPa for each of four temperatures, 200, 300, 400, and 500 K, involving the workflow tasks *pressure selection*, *NPT production*, and *isothermal analysis* of Fig. 1. The latter task enabled the calculation of the thermal compressibility coefficient and the two related properties, bulk modulus and sonic velocity. This was the last task of the workflow.

3 Discussion

In a nutshell, Table 1 gives the values of the PLGA(50:50) most relevant properties of the polymer-condensed phases obtained along the multiple tasks performed by our workflow. Calculation of the glass transition is not trivial. The calculated glass transition, T_g , was 330 K from the fluctuations of the enthalpy and 342 K from the Legendre transform of $H(T)$ illustrated in Fig. 2b, c, respectively. This T_g is higher than experimental values in the range 318–328 K [11]. However, the T_g bracketing temperature range we obtained, 262 ± 8 to 365 ± 11 shown in Fig. 2c, is consistent with measurements [12]. Most experiments are done for PLGA systems of molecular weight larger than the one used here. Comparing with values in the published literature, our results are in very good agreement, reinforcing the validity of our predicted results, which await experimental confirmation.

Table 1 Calculated PLGA(50:50) properties for a system with 12 chains with 222-mers each. The BCC atomic charges were used in the GAFF force field

	$T = 300$	$T = 400$	Experiment
Density ρ (kg/m ³)	1319.1 \pm 0.7	1288 \pm 1	1340 ^a
Heat capacity C_p (J/kg/K)	2977		
C_p (J/kg/K)	2921		
Thermal expansion α (10 ⁻⁴ K ⁻¹)	2.1 \pm 0.2		
Thermal compressibility κ_T (GPa ⁻¹)	0.14 \pm 0.01	0.21 \pm 0.01	
Bulk modulus B (GPa)	7.0 \pm 0.3	4.7 \pm 0.3	\approx 4 ^b
Sonic velocity s (m/s)	2318 \pm 39	1915 \pm 56	2326 – 2450 ^{c,d}
Cohesive energy/mer $E_{coh/mer}$ (kJ/mol)	19.13 \pm 0.06	18.23 \pm 0.08	
Hildebrand parameter δ_h (MPa ^{1/2})	19.69 \pm 0.03	18.98 \pm 0.04	

^a[7], ^b[8], ^c[9], ^d[10]

4 Conclusion

In this presentation we have described the numerous steps involved in modeling, simulating, and analyzing structural and thermodynamic response properties of poly(lactic-co-glycolic acid) (PLGA). Our protocol included exploration of a number of properties for the polymer under study, including the glass transition temperature, enthalpy, density, isobaric heat capacity, thermal expansion coefficient, isothermal compressibility, bulk modulus, sonic velocity, cohesive energy, and solubility parameter of PLGA. This approach is portable for the study of other polymers or complex materials.

Our findings are important in the design of nanostructures and nanoparticles as well as in the control of polymer folding when devising scaffoldings for tissue engineering. Part of this workflow was already successfully applied for analyzing the effects of solvents on the structure of solvated PLGA [5, 13]. Currently, we are extending the workflow use for the analysis of PLGA(50:50) with a variety of PLGA molecular weights [14].

Acknowledgments We acknowledge partial support from the Commonwealth of Virginia 4-VA grant no. 331050. JA is thankful for the Provost Office presidential scholarship support. All computations were done in Argo, the HPC platform of the Office for Research Computing, George Mason University.

References

1. M. Fathi, J. Barar, Perspective highlights on biodegradable polymeric nanosystems for targeted therapy of solid tumor. *Bioimpacts* **7**, 49–57 (2017)
2. Blanco, E., H. Shen, M. Ferrari, Principles of nanoparticle design for overcoming biological barriers to drug delivery. *Nat. Biotechnol.* **33**, 941–951 (2015)

3. T. Frisch, M.G., H. Schlegel, G. Scuseria, M. Robb, J. Cheeseman, G. Scalmani, V. Barone, G. Petersson, H. Nakatsuji, et al., *Gaussian 09, Revision D.01* (Gaussian, Inc., Wallingford, 2013)
4. D. Case, I. Ben-Shalom, S. Brozell, D. Cerutti, T. Cheatham, V.I. Cruzeiro, T. Darden, R. Duke, D. Ghoreishi, M. Gilson, et al., Amber18, 2018, University of California, San Francisco, CA
5. J. Andrews, E. Blaisten-Barojas, Exploring with Molecular Dynamics the structural fate of PLGA oligomers in various solvents. *J. Phys. Chem. B* **123**, 10233–10244 (2019)
6. GROMACS 2018 (2018). <http://www.gromacs.org>. Accessed 5 June 2020
7. M. Arnold, E. Gorman, L.J. Schieber, E.J. Munson, C. Berkland, NanoCipro encapsulation in monodisperse large porous PLGA microparticles. *J. Controlled Release* **121**, 100–109 (2007)
8. S.I.J. Wilberforce, S.M. Best, R.E. Cameron, A dynamic mechanical thermal analysis study of the viscoelastic properties and glass transition temperature behavior of bioresorbable polymer matrix nanocomposites. *Mater. Sci. Mater. Med.* **21**, 3085–3093 (2010)
9. H.-C. Wu, H.-H. Huang, C.-W. Lan, C.-H. Lin, F.-Y. Hsu, Y.-J. Wang, Studies on the microspheres comprised of reconstituted collagen and hydroxyapatite. *Biomaterials* **24**, 3871–3876 (2003)
10. N.G. Parker, M.L. Mather, S.P. Morgan, M.J.W. Povey, Longitudinal acoustic properties of poly (lactic acid) and poly (lactic-co-glycolic acid). *Biomed. Mater.* **5**, 055004 (2010)
11. Wypych, G., *Handbook of Polymers*, Elsevier, 2016.
12. T.A. Shmool, J.A. Zeitler, Insights into the structural dynamics of poly lactic-co-glycolic acid at terahertz frequencies. *Polymer Chem.* **10**, 352–361 (2019)
13. S. Hopkins, G. Gogovi, E. Weisel, R. Handler, E. Blaisten-Barojas, Polyacrylamide in glycerol solutions from an atomistic perspective of the energetics, structure, and dynamics. *AIP Adv.* **10**, 085011 (2020)
14. J. Andrews, R.A. Handler, E. Blaisten-Barojas, Structure, energetics and thermodynamics of PLGA condensed phases from molecular dynamics. *Polymer* **206**, 122903 (2020)

Part X
Grid, Cloud, & Cluster Computing –
Methodologies and Applications

The SURF System for Continuous Data and Applications Placement Across Clouds



Oded Shmueli  and Itai Shaked

1 Introduction

In many modern systems, data objects are replicated for survivability, latency reduction, and other reasons. Objects here may range from a bit; a byte; a file; a database of portions thereof; an XML element or document; a JSON document or portions thereof; a relational field, tuple, relation, or database; and so forth. In a hybrid cloud environment, as well as in a multi-cloud environment, an enterprise employs a number of local sites (or data centers) and cloud data center(s) that may be geographically distributed. The problem of where to place data and applications is complicated by (a) the variability in capabilities and APIs of the data centers of the various cloud service providers (CSPs), (b) the dynamic nature of demand for data and for applications, (c) pricing variability over time for various resources, and (d) enterprise business policies and regulatory constraints.

The placement problem touches on the following areas: the characteristics of applications using the data (relational or NoSQL database systems, ad hoc application programs viewing data as objects, and more), an optimization problem in a continuously changing environment, a distributed system problem in seamlessly moving data while providing an illusion of stationary data, and a recovery-type problem in handling failed data movements. It should be emphasized that the algorithms we present may be integrated into various system types. These systems

Supported by the Israel Innovation Authority, Kamin Project 57682.

O. Shmueli (✉) · I. Shaked
Computer Science Department, Haifa, Israel
e-mail: oshmu@cs.technion.ac.il

may have their own correctness requirements (e.g., ACID, eventual consistency) that they enforce.

The system we provide is charged with *managing the data placement and access layer*. This presents optimization challenges (that are not extensively discussed in this paper) that lead to placement decisions to be implemented. This paper focuses on the *implementation facet*, namely, algorithms that actually move data *proactively* from “old” places to “new” places. The challenges for this system include keeping track of where various data objects (often replicated) are located, implementing data movement decisions while minimally affecting the other systems and programs (called *clients*) using this data. Ideally, once a client gets a handle (pointer, address) on a data object (or replica), it can use it with no restrictions as far as our system is concerned.

We describe two types of algorithms: data movement (conservative and optimistic of various degrees) and recovery from various system faults. These algorithms are implemented on top of a ZooKeeper [5, 11] compact distributed database. The functionalities the SURF system provides may be extended to support concurrency control as well. The algorithms were tested within a prototype system that operates locally at the Technion as well as over three public clouds. The data movement algorithms strive for a minimal effect of data movement on clients’ operations. The system strives to create an *illusion* that no data movement takes place. There are, however, extreme circumstances in which a client’s request may be refused (or aborted from our system’s point of view). The client may retry later on or actually abort its execution.

Let us briefly touch on the optimization aspects. They rely on the mathematical programming sub-area of *goal programming* [10, 14] that is tailored to *multi-objective* optimization. Data objects’ granularities are an enterprise’s choice, and the optimization technology is agnostic to this aspect. We employ the concept of a *business goal*. A business goal is a mathematical object that is associated with an *individual* object. A business goal states, for various attributes, their level of importance on a scale from 1 to 10. For example, attributes may refer to security (S), speed of access usability (U), replication (R), and more. No generality is lost as these attributes are further weighted, describing their importance relative to each other.

Next, system performance parameters are taken into account as well as forecasts of future system characteristics. The end result is a formulation of a quadratic *goal program GP*. The goal program is a mathematical program that “encodes” the objects’ business goals, pricing tables of the various CSPs (including nonlinear ones), and budgetary and placement constraints. The program presents an overall *goal G* to be minimized. In fact, *G* expresses the overall enterprise business policies as well as the environmental realities. The solution to *GP* assigns values for decision variables (indicating locations for data objects) that minimize *G* subject to *GP*’s constraints.

For a large number of objects, solving *GP* is intractable. We use an approximation technique that assigns objects to *groups*. If *o* and *o1* are objects in the same group *g*, then all their replicas reside precisely at the same sites. The number

of groups is a changeable system parameter. The assignment of objects to groups depends on values for the attributes describing the object. Roughly, similarly “appearing” and “behaving” objects are assigned to the same group. The assignment to groups may be altered once a data movement decision is reached. Thus, o and $o1$ may belong to different groups, say $g1$ and $g2$, following a data movement. The main task of the data movement algorithms is to *manage the process of changing from one configuration of objects to groups to sites to another while disrupting minimally data usage*. These algorithms, as well as the system’s directory services, are implemented on top of a small Zookeeper [5, 11] distributed database. These algorithms can also place various types of applications to sites, a functionality that will not be described herein.

The system may support the following types of data systems: (1) A *consultant* decision support system that recommends where each data item should be stored. The system may also be used to examine various *what-if* scenarios. (2) A *broker* is a system that has the aforementioned decision components. The broker exposes an interface similar to that of S3 [15], for storing and retrieving objects; objects are stored in one or more data centers and local sites. The broker enables object-oriented applications that utilize it for information storage and retrieval. SURF’s distributed algorithms enable data movements while applications are running. (3) Data systems that handle their own data access, concurrency control, and replication may use a *comprehensive system* to perform actual data access. These data systems perceive a static universe although data location may seamlessly shift during their operation. (4) Finally, the comprehensive system may be embedded in a *single data system* (e.g., a relational database system or a NoSQL system) within its storage handling layer, thereby providing dynamic, optimized data placement maintenance. In system types (2), (3), and (4), user consent may be requested prior to carrying out data relocation movement. SURF’s data distribution decisions technology was examined, over a number of public clouds, via extensive simulations on a hypothetical commerce application and was found to be superior, in a changing environment, to naive placement methods.

The rest of the paper is organized as follows. Section 2 presents basic distributed algorithms that enable data movement (to achieve optimized placement), while applications read and write data items. The algorithms utilize a placement and locking data structure, implemented via ZooKeeper as described in Sect. 3. Section 4 describes mechanisms for recovery of the SURF system under various failure scenarios. Section 5 reviews related work. We conclude in Sect. 6.

2 Data Movements Architecture and Algorithms

2.1 Background

Recall that the main task of the data movement algorithms is to *manage the process of changing from one configuration of objects to groups to sites to another while disrupting minimally data usage*. We shall use the term “tuple” to represent an object. This is especially appropriate as we represent information in relational tables, and each object usually has at least one tuple representing it. Of course, we can represent objects in other ways (e.g., as a record in a NoSQL data store or even by ordinary files); however, representation via tuples is convenient and easy to explain. We shall therefore often blur the distinction between the object and the tuple representing it.

A basic problem that has been extensively researched is that of concurrency control over replicated objects. See Sect. 5. However, the problem of object (tuple) migration has not been clearly addressed. This problem deals with the scenario in which the location of tuples needs be changed while the system and applications continue operations. This problem is becoming very relevant in the context of the cloud and its continuously changing pricing, performance, and security classification of providers and data centers. In such an environment, changing data placement is an ongoing decision-making process. However, the technical challenge that we tackle is how to move objects (tuples) with only minimally disrupting ongoing activity that uses the moved data.

2.2 Data Structure

The main data structure for all three algorithms is a tuple directory table, shown in Table 1, with columns TupleID, OddGroup, OddLocations, EvenGroup, EvenLocations, Locked, InEffect, AccessL, Tampered, and Waiting.

Consider the first line of Table 1. The tuple ID is fixed and it is 1217. The effective state of this tuple is *odd*, as indicated in column InEffect. This tuple is not locked (Locked is *none*). The group in effect is group 16, as indicated in column OddGroup. This group 16 should be thought of as *16-odd*. As such, a copy of this tuple is stored in locations L1, C1, and C2, as indicated in column OddLocations. Similarly, tuple 1218 is in group 5, and as the InEffect value is *even*, the tuple should be thought of as being in group *5-even*. Copies of tuple 1218 are therefore stored in locations L2, C2, C4, and C5, as indicated in column EvenLocations. Tuple 1218 is currently being moved, as indicated by the value *move* in the Locked column. Whenever a tuple is currently *even* (respectively, *odd*, as indicated in column InEffect), and it is being moved, once the move is complete, the tuple will become *odd* (respectively, *even*, as indicated in column InEffect).

Column AccessL indicates whether the tuple is being accessed, and if so which copy or copies are being read and by which process. In the example shown in Table 1, tuple 1218 is currently being read by process 2434 in location C1 and by process 2409 in location C2. A value of *none* in this column indicates no process is currently reading the tuple.

Column Tampered indicates whether and which process (or processes) changed any copy of the tuple. In our example, tuple 1220 was changed by process 2000. A value of *none* in this column indicates the tuple is not being updated. The function of this column will be further explained in the context of the optimistic algorithms.

A global variable EvenOrOdd, containing either *even* or *odd*, is used to indicate whether globally the last completed move established InEffect = *even* or InEffect = *odd*. Initially it is set to *odd*. We note that this allows us to keep the even or odd locations at the group level, rather than for each tuple individually. Table 2 shows an example of such a table. In this example, if globally *odd* is in effect, group *1-odd* tuples are in locations L1, C1, and C2; and if *even* is in effect, group *1-even* tuples are in locations L2, C2, and C3. Using Table 2 allows us to eliminate columns OddLocations and EvenLocations from Table 1, as the same information is conveyed by Table 2 in succinct form.

2.3 The Order of Moving Tuples

Regardless of the algorithm used to move tuples, the *order* of moving the tuples, from current groups and their locations to new groups and their locations, may be significant. Intuitively, the move may include some very beneficial tuple moves that should be executed as soon as possible. The system can estimate the immediate benefits (to both business goals and budget) for moving any particular current group G . The one group G_{i1} such that moving its tuples is expected to yield the highest benefit (in terms of business goal, budget ,or some combination thereof, as set by the system’s operator) is moved first so that these benefits are realized quickly. Once G_{i1} is moved, one can consider the other groups in the new setting and choose the next one to move, say G_{i2} , which will likely yield the highest benefits. This process is repeated until the tuples of all groups are moved.

Table 2 Group-level location table

Group	OddLocations	EvenLocations
1	L1, C1, C2	L2, C2, C3
2	C2, C4	C1, C5
...

2.4 Pessimistic Move Algorithm

We detail the steps of performing a data movement. The description presents a sequential algorithm. In fact, tuples may be moved in parallel with some modifications to algorithm 9. Further optimizations are possible, and we elude to some of them 21:

1. Let S_0 denote the set of currently active processes in the system at the moment of time T_0 when the decision to move tuples is reached.
2. Once a decision to move is reached, S_0 is recorded.
3. During a move, all newly joining processes are notified “start of moving” as soon as they join the system.
4. Once “start of moving” is received by a process, each new access *needs* an *A*-mode lock, whether the access is a read or a write access. *A*-mode locks are compatible with other *A*-mode locks.
5. Once all processes in S_0 have terminated (say at time T_1), we are ensured that all active processes have been setting locks in column Locked for all of their accesses, and we then start the actual moving of tuples. The interval $(T_1 - T_0)$ is expected to be relatively short (a few minutes). In case of a very long (as defined by a system parameter) running process, these processes may be aborted, unless overridden by a system operator (in which case the move is either aborted or delayed).
6. The move is performed in sets of tuples, treated one by one. Usually each such set is taken out of a particular group of tuples among the groups that are currently in effect. A tuple set may constitute the whole group or a proper sub-group of the group. See below (21) for various sub-group formation options. It is also possible to form a set out of tuples in several groups; we shall not discuss this option further, as the algorithm for this case is nearly identical to the one described here.
7. Prior to moving a tuple, the move process needs to lock it in *M*-mode.
8. Consider the j^{th} sub-group SG_{ij} of tuples in group G_i that need be moved out of its current group (say, *N-even*) into k new groups (say in $M_1\text{-odd}, \dots, M_k\text{-odd}$).
9. Before moving the sub-group tuples, they must be locked in *M*-mode by setting the value in column Locked to *move* (*none* means not locked). *M*-mode lock is *incompatible* with *A*-mode locks.
10. If the lock is already set (to *A*-mode, as there are no concurrent move processes), the move process *waits* for this lock to become *none* (it was set by an application that is still using the tuple, namely, one or more of its copies). Otherwise, the tuple is locked by the move process in *M*-mode. The possibility of the move process proceeding to handle other tuples when a locked tuple is encountered will raise the possibility of livelocks and will necessitate livelock detection and resolution or prevention (can be done by standard techniques). Alternatively, the move process can simply *wait* until the tuple is unlocked (Locked is *none*). We choose this option in this basic algorithm.

11. To avoid deadlocks, if an application tries to lock a tuple in *A*-mode that is either locked in *M*-mode or for which the move process is waiting to be unlocked, the application *aborts* (i.e., the access is refused and results in an error) and may retry later on. Therefore, the move process may wait for an application, but an application never waits for the move process; hence the move process is never involved in a deadlock cycle. The waiting is indicated in the column *Waiting*, where 0 means no waiting and 1 means waiting.
12. While locked in *M*-mode, a tuple cannot be accessed by applications or systems, as *A*-mode locks are incompatible with *M*-mode locks.
13. New tuple copies are established in new locations as needed, while existing copies are left as is in existing “old” locations. If an “old” location is also a new location, the “old” copy may be used instead of creating a brand new “new” copy. As described, any tuple replica may be the one whose content is copied to new locations. However, other policy variants are possible. For example, examine the current replicas and choose the one with the latest update timestamp as the source for copying or use each current replica as a source for one new replica (and if the number of replicas is changed, use a heuristic).
14. Once all moved sub-group tuple copies are established, *InEffect* is set to *odd* (respectively, *even*) if originally *even* (respectively, *odd*), the sub-group tuples are then *unlocked* (Locked is set to *none*).
15. As already indicated, we need not keep the locations for each tuple; instead, we can store the locations per group for each of *odd* and *even*, in a separate table (Table 2), as the current locations are indicated in column *InEffect*.
16. Once *unlocked*, a tuple may be accessed by processes where the locations are determined according to column *InEffect*.
17. Copies of tuples that now occupy space in not in effect locations may be discarded.
18. When the size of the sub-group being moved is large, this may freeze some system operations, namely, those operations that need to access a locked tuple. However, freezing usually lasts for a short duration.
19. Observe that during the move, some tuples in a group may belong to their “old” groups, while others are already in their “new” groups.
20. To make unavailability periods short, tuples may be moved in small sub-groups (an extreme case is sub-groups consisting of a single tuple). This has the drawback that one may be unable to ship a large group of tuples together to a new location.
21. A sub-group may be determined in a number of ways:
 - (a) Sequentially, in a sort order or physical proximity order within the current group.
 - (b) Alternatively, based on membership in a new group. For example, move tuples from a group of type *odd* that are destined to “new” group 23-*even*.
 - (c) Some other way, e.g., a combination of the above approaches.
22. When all tuples of old groups are handled, the new groups become the current groups, and the global variable *InEffect* is toggled from *even* to *odd* or vice

versa. At this point information regarding old groups may be deleted. In addition, “end of move” is sent to all active processes, so that they may resume normal operation without *any* locking, as well as releasing their current locks in Table 1.

Note that this algorithm ignores lock columns other than Locked—these columns may be needed in order to support various algorithms such as replica updating algorithms and transactions using replicas. These may be implemented via additional columns that enable synchronization functionalities, in addition to the existing lock columns.

We observe that the pessimistic algorithm experiences no deadlocks or livelocks. It reclaims the entire “old” space. It may be delayed by processes holding *A*-mode locks for long durations (a standard mechanism for detecting and aborting such processes may be employed).

2.5 *Optimistic Algorithm 1*

We now present the first optimistic extension to the pessimistic move algorithm described above. This extension allows read accesses to tuples, but not write accesses, during the move. Intuitively, after performing a move, if there are still active readers, certain “old” tuples’ space may not be reclaimed. This means that the whole move may be accomplished with some leftover old space that may be reclaimed later on. The general outline of this algorithm is similar to that of the pessimistic algorithm described in Sect. 2.4, and so we shall only list those steps in which the algorithms differ, where all steps not explicitly listed below should be performed as in the previous algorithm.

4. Once “start of moving” is received by a process, each new access *needs* an *R*-mode or a *W*-mode lock, depending on whether the access is a read or a write access. *R*-mode and *W*-mode locks are compatible with other *R*-mode and *W*-mode locks.
9. Before moving the sub-group tuples, they must be locked in *M*-mode by setting the value in column Locked to *move* (*none* means not locked). *R*-mode locks are compatible with *M*-mode locks (as well as with *W*-mode locks and other *R*-mode locks), while *W*-mode locks are *incompatible* with *M*-mode locks (but are still compatible with *R*-mode locks as well as other *W*-mode locks). If a process attempts to obtain a *W*-mode lock while there is either an *M*-mode lock set or the move process is waiting for an *M*-mode lock, the attempting process is *aborted*.
- 10a. If the lock is already set (to *W*-mode, as there are no concurrent move processes, and *R*-mode locks are compatible with *M*-mode locks), the move process waits for the *W*-mode lock to be cleared (it was set by an application that is still using the tuple, namely, one or more of its copies). Otherwise, the tuple is locked by the move process in *M*-mode.

- 10b. When reading a copy, a process obtains an R -mode lock on the tuple, and the site S in which the copy is accessed is recorded in column AccessL , together with the process ID; note that there may be multiple concurrent readers on one or more copies of a tuple. When a reading process finishes accessing a tuple, its entry in AccessL is discarded. If subsequently AccessL is empty, it is set to *none*, and the R -mode lock may then be discarded. Essentially, this means an R -mode lock is indicated by AccessL having a value other than *none*. This implies one cannot discard the space occupied by the copy of a tuple at site S (that is strictly an “old” site for this tuple) if at the end of the move there is an R -mode lock on the tuple at site S (as recorded in AccessL).
11. This step is the same as in the pessimistic algorithm; only here W -mode stands for the previously mentioned A -mode.
12. While locked in M -mode, a tuple cannot be written by applications or systems, as W -mode locks are incompatible with M -mode locks. Tuples may still be read while locked in M -mode, as R -mode locks are compatible with all lock modes.
14. Once all moved sub-group tuple copies are established, the M -mode lock on the tuple may be removed as soon as there is no R -mode lock set with ongoing accesses (as indicated by AccessL) in strictly old locations. This is necessary in order to prevent two distinct versions of the same data item to co-exist at the same time where one evolves and one is static. Once the move lock is removed, InEffect is set to *odd* (respectively, *even*) if originally *even* (respectively, *odd*), the tuple is then unlocked (Locked is set to *none*). Observe that had the M -mode lock been removed with R -mode set in a strictly old location, this would enable writes which would take place only in the new locations, creating two distinct versions; this can be fixed by having writes also update old locations; however this may further complicate *this* algorithm and algorithms which may use it as a service. Once all sub-group tuples are unlocked, we can process the next sub-group.

We observe that optimistic algorithm 1 experiences no deadlocks or livelocks. Further, assuming reads eventually terminate, it will reclaim the entire “old” space. It may be indefinitely delayed by long-running readers (a standard mechanism for detecting and aborting such readers may be employed).

We now discuss the reasons for not removing the M -mode lock while active readers are present. For example, it is possible that the move starts with $\text{AccessL} = \{(P_i, S_x)\}$ (with i being some process ID and S_x being some site) and $\text{Locked} = \{\text{move}, R\}$. During the move, the AccessL column is reset to *none*, and subsequently Locked is set to $\{\text{move}\}$. Later on, other accesses reset Locked to $\{\text{move}, R\}$ as $\text{AccessL} = \{(P_j, S_y), (P_k, S_z)\}$, where S_y and S_z are strictly old sites. This means that when the move of the sub-group of this tuple completes, copies on sites S_y, S_z cannot be discarded, and their space may not be reclaimed. Once AccessL contains no strictly old locations, the space at S_y and S_z may be reclaimed. Theoretically, such space may be indefinitely prevented from being reclaimed in this case; the system may alert an operator if the length of the reclaiming prevention period

exceeds a system parameter. Another option is aborting operations which hold an entry in the AccessL column during the period between two consecutive tuple moves. If we allow updating the tuple once the move is complete but there are still old copies at S_y and S_z , we will need to update these old copies as well. In order to reduce the possibility of such costly extra updates, we choose not to allow such tuple updates until AccessL becomes empty (at which point the M -mode lock is removed).

2.5.1 Optimistic Algorithm 1.1: Smart Waiting

While Optimistic Algorithm 1 allows applications to read tuples while they are being moved, it requires any application attempting to write to a tuple that is being moved or for which the move process is waiting to immediately abort the access. We now present a refinement of Optimistic Algorithm 1, in which applications are much less likely to abort yet is still livelock *and* deadlock-free. This refinement relies on the observation that since W -mode locks are compatible with both R - and W -mode locks, the only possible deadlock cycles consist of the moving application and a **single** writing application (i.e., an application explicitly attempting to obtain a W -mode lock).

Optimistic Algorithm 1.1 is therefore identical to Optimistic Algorithm 1, except for step 9 which is changed accordingly:

9. This step is the same as in Optimistic Algorithm 1 with regard to both R -mode and M -mode locks. If a process attempts to obtain a W -mode lock, it must do the following:
 - (a) If there is either an M -mode lock set or the move process is waiting for an M -mode lock on the tuple, the process *waits* until the M -mode lock is released before obtaining the W -mode lock. If there is no M -mode lock currently held and the move process is not currently waiting for one on the tuples, the process obtains the W -mode lock and proceeds as in Optimistic Algorithm 1.
 - (b) While *waiting*, the process continuously monitors all tuples for which it holds W -mode locks for the existence of any pending M -mode lock.
 - (c) If at any point the move process attempts to obtain, or starts waiting for, an M -mode lock on such a tuple, the process is immediately *aborted*. Since W -mode locks are only incompatible with M -mode locks, it is guaranteed that either the current M -mode lock will be cleared, or else the process will be forced to abort.

We observe that Optimistic Algorithm 1.1 experiences no deadlocks or livelocks, and applications are aborted only when necessary in order to avoid deadlocks. Further, assuming reads eventually terminate, it will reclaim the entire “old” space. It may be indefinitely delayed by long-running readers (again, a standard mechanism for detecting and aborting such readers may be employed).

2.6 *Optimistic Algorithm 2*

There is a second optimistic algorithm, which extends Optimistic Algorithm 1, allowing writes while a tuple is being moved. In this extension, a move may need to be repeated if old copies are updated. Therefore, Optimistic Algorithm 2 needs a mechanism for testing for this condition. In addition, there is no point in moving a tuple while it is being updated. Due to space limitations, we shall not elaborate further on this algorithm.

3 Data Movement Algorithm Implementation

In this section we describe in detail the implementation of one variant of the data movement algorithms using the ZooKeeper distributed database system. The algorithm chosen is Optimistic Algorithm 1.1 (see Sect. 2.5.1), although the mechanisms described can be easily adapted to implement any of the other algorithms discussed in the previous section.

The implementation was tested in various scenarios, across geographically separated servers, using simulated data access.

3.1 *ZooKeeper Data Structure*

ZooKeeper [5] provides a hierarchical key-value store, with the keys referred to as *znodes*. Each *znode* is created in either *persistent* mode, meaning the *znode* persists until explicitly deleted, or *ephemeral* mode, meaning it will be automatically removed whenever the session in which it was created expires. Furthermore, ZooKeeper allows creating *sequential znodes*—such nodes are assigned a *unique* sequence number within their hierarchy, allowing for consistent ordering.

Since ZooKeeper is schemaless, we need to define how the data of Table 1 is represented. Note that ZooKeeper limits node data to 1MB, and so most of the information can be thought of as residing in the hierarchical structure itself. In fact, the implementation described henceforth only calls for reading actual node data in cases of long-running or misbehaving applications, meaning that during normal operation applications need only query the *structure* of the tree, allowing for faster operation.

Figure 1 shows the general schema for the data structure. The hierarchy consists of three main branches—*tuples*, where information about individual data tuples and their lock status is stored; *apps*, where information about currently running applications is stored; and *migration* which is used for monitoring the data movement process.

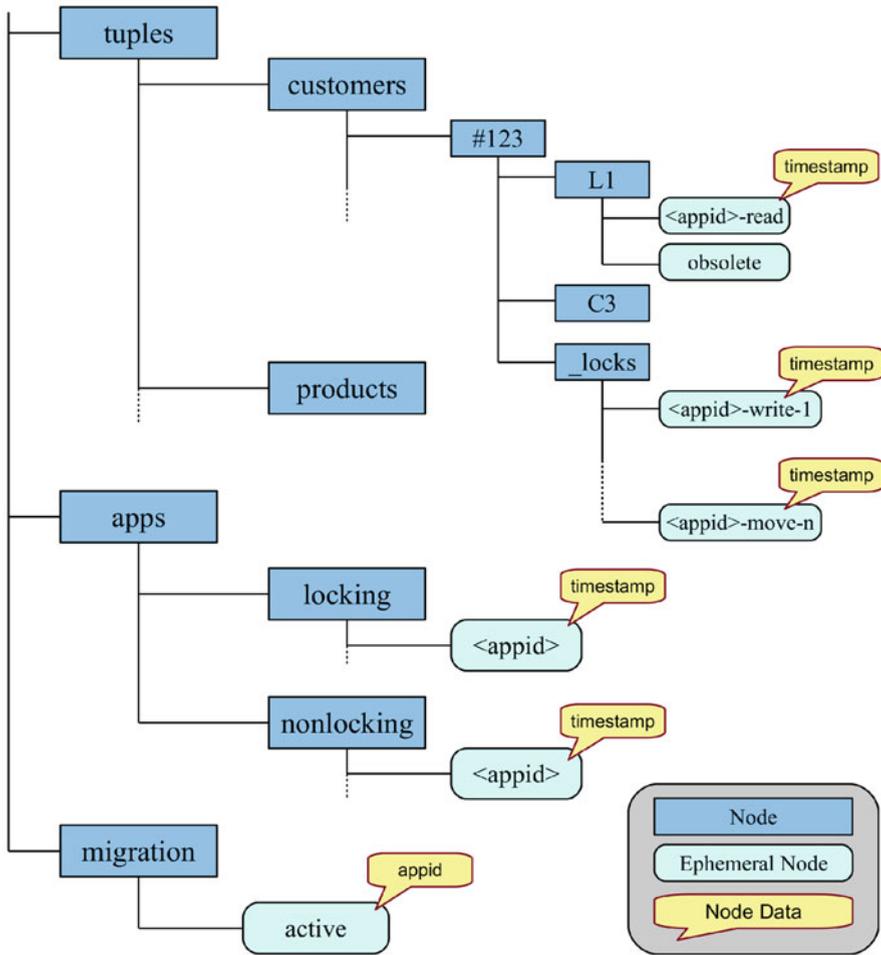


Fig. 1 ZooKeeper data structure for Optimistic Move Algorithm 1.1

Data in the *tuples* branch is organized according to tuple type. In this example we assume the applications are those of an online store system, where the different tuples may be divided into types such as customers, products, etc. Within each branch there is additionally a branch for each concrete tuple of that type. Concrete tuple branches (e.g., /tuples/customers/#123) contain nodes for each site on which the tuple is currently replicated. When no movement is taking place, these nodes contain no child nodes; however, during migration each such node may contain a child node named *obsolete*, meaning this location is obsolete for this tuple, and the space taken by it is designated to be reclaimed. The specific location nodes are also where *R*-mode locks are logged, as *R*-mode locks are always tied to a specific location (similar to column AccessL in Table 1).

Each concrete tuple branch also contains a node named *_locks*, which serves to record and maintain *W*-mode and *M*-mode locks, as such locks apply to all replicas of a tuple (similar to column Locked in Table 1).

The *apps* branch is used to monitor all currently running applications as well as their status regarding the movement process. This information is used by the movement process to make sure all currently running applications have been notified “start of moving.”

Finally, the *migration* branch is used to notify the start and end of the (data) movement process. Specifically, the existence of a node named *active* under *migration* is the equivalent of notifying any newly starting application that a migration is taking place.

3.2 Locking Implementation

3.2.1 Movement Signaling

While no movement is taking place, there is no need for locking, and so applications should not waste time unnecessarily locking and releasing tuples. When a movement begins however, we must have a way of making sure all applications respect the locking protocols. For this reason, upon starting, every application must check for the existence of a node named *active* under the *migration* branch—if such a node exists, the application starts in a *locking* mode, and otherwise it starts in *non-locking* mode. To avoid a race condition, each application performs the following steps, where $\langle \text{appid} \rangle$ stands for the application’s process ID:

1. Construct a *transaction* consisting of the commands [create /migration/active; create ephemeral /apps/non-locking/⟨appid⟩ with current timestamp as the data; delete /migration/active]. The create-delete pair takes advantage of ZooKeeper’s transaction atomicity.
2. Attempt to execute the transaction. If it succeeds—start in *non-locking* mode—under which the application is exempt from following the locking protocols.
3. If the transaction fails, create an ephemeral node /apps/locking/⟨appid⟩ with the current timestamp as the data. The application then starts in *locking* mode and must follow the locking protocols. Upon application end, the node is removed.

This mechanism ensures no application will erroneously start in a non-locking mode after a move process has signaled a movement is about to start. Note that since the node is created in *ephemeral* mode, it will be automatically removed in cases of a complete failure of the system hosting the application.

When the decision to move is reached, the move process initializes and creates an *ephemeral* node named *active* under the *migration* branch. The move process then sets a watch on the *non-locking* sub-branch of the *apps* branch, waiting until it has no children—this is equivalent to recording the set of active applications S_0 described in the previous section.

3.2.2 Read-Mode Locks

We assume the majority of data access will be exclusively read, and so read locks must have as little overhead as possible while making sure no application attempts to read a tuple from an obsolete location and no replicas are deleted while still being accessed. To achieve this, during data migration, an application wishing to read a tuple performs the following steps:

1. Choose an appropriate site for reading. The logic used to make the choice is inconsequential to this algorithm. Denote the corresponding path by P , e.g., if the location $L1$ was chosen for reading “customer#123,” set $P=/tuples/customers/#123/L1$.
2. Construct a *transaction* consisting of the commands [create $P/obsolete$; create ephemeral $P/read-<appid>$ with the current timestamp as the data; delete $P/obsolete$].
3. Attempt to execute the transaction. If it succeeds—proceed reading from the selected location. Else, if the transaction fails, return to step 1 choosing a different location.
4. When done reading, remove the node $P/read-<appid>$.

Since the reading application employs a *transaction*, either all of the commands in 2 execute or none at all. This means if the transaction succeeds there will be no node named *obsolete* under P , but there will be a *read* node for the application. Transaction atomicity means reads are indeed compatible with other reads, as no other client will see the transient *obsolete* node.

3.2.3 Write-Mode Locks

Write-mode locks are to be compatible with other write-mode locks, but not with move-mode locks. As there is only one movement process active at any point in time, this can be achieved by a standard reader-writer lock, with a slight change in terminology—here the shared locks are *write* locks, while the exclusive lock is the *move* lock.

Explicitly, when an applications wishes to write to or update a tuple while movement is active, it takes the following steps:

1. Create an ephemeral *sequential* node named $<appid>-write-$ under the *_locks* node of the appropriate tuple, with the current timestamp as the data. Note that when creating a *sequential* node ZooKeeper will append the sequence number to the end of the given node name.
2. Read the children of the *_locks* node of the appropriate tuple—since this is done after creating a sequential node under the *_locks* node, the application is guaranteed to see any children having a lower sequence numbers than the one it was assigned.

3. If there are no children with lower sequence having “-move-” in their name, the application acquires the lock and may proceed updating the tuple. Note the application is responsible for updating all, or some of, the current locations, in accordance with its correctness constraints.
4. Otherwise, if there is a move lock with a lower sequence number:
 - (a) The application sets a watch on the move-mode lock node—since there are no concurrent move processes, there will be at most one such node for any given tuple.
 - (b) Read the siblings of any write-lock node this application currently holds, setting watches for each of them. That is, the children of a *_locks* node are queried whenever this application has created and not yet deleted a write-lock node under it.
 - (c) If any of the currently held write-lock nodes have move-lock siblings, the application immediately aborts, releasing all locks it may hold (either read or write). Alternatively, if the move-lock node holding back the current write-lock is released—the application acquires the write lock and may proceed.
 - (d) Otherwise, the application waits for either of the watches to trigger and then returns to step 4.

As explained in Sect. 2.5.1, the algorithm described in step 4 ensures the application aborts if it may cause a deadlock while allowing it to wait as long as it provably cannot be a part of a deadlock cycle.

3.2.4 Move-Mode Locks

Move-mode locks are created by the move process alone and represent the exclusive part of the shared-exclusive mechanism (the write-mode locks being the shared part as described above). When the move process decides, a concrete tuple should be relocated, that is, either new copies of it created, current copies removed, or a combination of both, it does the following:

1. Create an ephemeral *sequential* node named <appid>-move- under the *_locks* node of the appropriate tuple, with the current timestamp as the data. Note that when creating a *sequential* node ZooKeeper will append the sequence number to the end of the given node name.
2. Read the children of the *_locks* node of the appropriate tuple—since this is done after creating a sequential node under the *_locks* node, the application is guaranteed to see any children having a lower sequence than the one it was assigned. Since there may only be one move process operating at a time, any nodes other than the one this process created must be write-lock nodes.
3. If there are no children with a lower sequence number, the move-mode lock is acquired, and the move process may proceed to create additional replicas of the tuple as needed.

4. Otherwise, if there are any children with a lower sequence number, a watch is set on the node having the lowest sequence number. The process then waits for the watch to trigger, returning to step 2 when it does.

3.3 Reclaiming Space

Clearly, space occupied by tuples that are no longer accessible to clients' processes need be reclaimed. Space reclamation may be a separate step or be performed as part of the move process. Due to space limitations, we shall not elaborate on the details of space reclamation.

4 Handling Failures

The SURF system is a distributed system that operates over a number of sites. Its data is distributed over these sites and replicated. ZooKeeper handles directory services and locking services (while data is being relocated). ZK is itself a distributed (small) database system with a tree-based data model. This means that SURF has a number of failure possibilities that need be addressed.

Depending on the mode of deployment (i.e., consultant, storage layer, full system, etc.), SURF may either fully control, be a part of, or even be completely ignorant of concurrent tuple modifications, which may result in the same logical tuple having different replicas. This may be allowed or even required by the client's systems, but it presents challenges regarding data restoration and recovery. Therefore, whenever SURF does not fully control tuple access and concurrent modifications, the client must specify for each tuple or group of tuples the desired *recovery policy*, choosing one of the following basic options (the system may be extended by users to other options):

1. Latest-version recovery: in case of failure, tuple data will be restored to the latest known valid version (with a consistent ordering of sites used as a tiebreaker in case of distinct versions having the same timestamp). Note that in case of a prolonged network failure this policy may result in tuple data being overwritten with a newer version where the client application may not have meant for it to be overwritten.
2. Majority recovery: in case of failure, tuple data will be restored to the version held by the majority of known valid replicas. If no such majority exists, either because more than half the sites holding replicas of this tuple failed or because no single version is held by a majority of the surviving sites, the tuple is restored according to the latest-version policy. Site ordering on a per-tuple basis for some tuples is also possible.

3. Site ordering in which a total order on sites determines which version has priority. For example, if a tuple is on sites i and j and $i < j$, then i 's tuple version if available is used rather than j 's version.
4. Application ordering: similar to site ordering but based on priority among applications. Application ordering and site ordering may be combined. Application ordering on a per-tuple basis for some tuples is also possible.

We distinguish between two types of failures: failure of a ZK instance and failure of a site on which a copy of a tuple resides. It is possible that a site failure will also cause a ZK instance to fail. However, we distinguish between these two kinds of failures, as their handling is different. To be able to handle potential failure, SURF invests during normal operation in creating an infrastructure that would enable SURF's recovery. This infrastructure consists of a directory and files (both replicated) that enable creating a consistent ZK worldview and files that enable creating tuples' versions on a best-effort basis when all other ways fail.

4.1 Identifying Versions

During recovery SURF may need to pick between different versions of the same tuple or even identify that two copies are distinct. Since tuples may be arbitrarily large, it is impractical to transfer the entire data in order to determine whether copies are identical or not. To overcome this, the local data access mechanism at each site must provide a way to query the timestamp (or version ID) of any local copy, as well as the hash of its data. It is inconsequential to SURF which hashing algorithm is used, so long as all data sites use the same algorithm (e.g., MD5 or SHA1 may be used). The user should choose an algorithm which balances speed with accuracy. Some data storage facilities may have a built-in hashing mechanism which may be used to avoid calculations during recovery (when it is crucial to resume operation quickly). SURF's Connector component allows the local site to query each data site for the timestamp and hash of every local copy of a tuple, so that the data transferred is minimal and independent of tuple sizes. In the case of the latest version, site ordering and application ordering, we first find the canonical version by selecting the first copy in the appropriate order (querying possible locations for timestamp as needed). The hash of every copy is then queried, and we keep only those copies whose hash matches that of the canonical version. The case of majority recovery is slightly different, as we cannot rely on any ordering of the different copies. Instead, the hash of every copy is queried, and the copies are divided into groups having equal hash. The copies kept are those in the largest group, with ties broken according to the timestamp of the latest copy in each group.

4.2 Recovery Algorithm Description

4.2.1 Normal Operation

1. On its initial installation, all data is located locally, and a detailed map of tuple locations is kept in a file `map0`. Before SURF starts its normal operation, `map0` is securely saved in secondary storage in `k0` (a parameter) failure-independent locations. The locations of the file `map0` are kept in a file `loc0` in all local locations; `loc0` also stores the current SURF time. SURF then does a “dummy” data movement and records its success by creating, in all operational sites, a copy of `loc0` called `loc1` as well as files `success0` and `success1` recording the current time. The directory in which `map0`, `loc0`, `loc1`, `success0`, and `success1` are stored is always called *recovery*. SURF begins normal operation once `map0`, `loc0`, `map1`, `loc1`, `success0`, and `success1` are securely stored.
2. During operation, if the j th decision to relocate data is reached, SURF forms a file `mapj` of the planned locations and starts data movement only once `mapj` is securely saved in secondary storage in K failure-independent locations ($1 \leq K \leq |S|$ is a parameter that may be reset by the operator, $|S|$ is the number of sites). The location of the file `mapj` is kept in a file `locj` in all operational local locations and all operational data centers; `locj` also contains the current SURF time. SURF starts actual data movement once `mapj` and `locj` are securely saved in the recovery directories.
3. SURF starts actual data movement only if all the sites are operational at the start of moving and acknowledge writing the new `mapj`. This means that data movement may be delayed for a long time in case of a prolonged site failure. On the other hand, this ensures a consistent view of all sites as to when data movement takes place. To handle the case of a prolonged site failure, we may remove this failing site from the system altogether. When it recovers, we will consider it a new site. It will rejoin during a new data movement operation. Its joining is expressed in that new locations for tuples will also be located in this new joining site. In addition, new data movement may be forced on SURF.
4. At the successful end of the j th data movement, SURF writes a short file named `successj` in all operational sites. The file contains j and the time (i.e., timestamp) of data movement completion as determined by SURF. The existence of `successj` is the indication that the j th data movement was completed.

4.2.2 Partial ZooKeeper Failures

In this scenario, some sites on which ZK instances are running have failed. This means:

1. A new ZK instance is brought up.
2. ZooKeeper brings this instance up to date.
3. If sufficiently ZK instances are not yet operational, ZK goes back to step 1.

4. SURF continues normal operation throughout this process.

4.2.3 Total ZooKeeper Failure

In this scenario, all ZK instances have failed. This implies that SURF (once up) can get no access to the enterprise's data. This means SURF needs to bring ZK up, either by itself or via operator (human) services. Once ZK is up and running, it needs be informed where data is located. For this, we employ the following mechanism:

1. When recovering from total ZK failure, SURF examines file loc_0 in any operational site. This provides SURF with the location of directory recovery in all sites. SURF then obtains, from all relevant operational sites, the two latest versions of *success*, say $success_i$ and $success_{i+1}$ and of *loc*, say loc_j and loc_{j+1} . Using the *loc* files, SURF obtains also the two latest versions of *map*, say map_j and map_{j+1} . We consider the following cases:

- (a) $i = j$. This means SURF successfully completed its j th data movement. Therefore, the up-to-date data locations may be obtained from any copy of map_j . This also means there was no ongoing data movement operation when ZK totally failed (due to the consistency requirement and acknowledgment that all sites are operational at start of data movement). Thus, SURF may simply continue operating as usual, and the recovery process is complete. It is possible that prior to ZK total failure, ZK lock tables were used by systems (e.g., a database system) and/or applications to synchronize activities. The failure might have left data in an inconsistent state from the point of view of these systems. SURF cannot guarantee recovery to a consistent state from the point of view of these systems and/or applications. Therefore, it is the responsibility of these systems and/or applications to maintain their own recovery data. Once ZK is up and running, they may examine recovery data, bring their own states to a consistent state, and proceed normally once this is accomplished. It is possible that such recovery data itself is maintained as SURF tuples. It is also possible that other mechanisms (outside the scope of control of SURF) are used for recovery purposes.
- (b) $i = j - 1$. This means SURF started its j th data movement, but has not completed it. The implications for tuple locations are as follows:
 - (i) Some tuples may be only located in their old locations, i.e., as indicated in map_{j-1} .
 - (ii) Some tuples may only be located in their new locations, i.e., as indicated in map_j .
 - (iii) Some tuples (e.g., those accessed by long duration readers) may be located in locations indicated by both map_{j-1} and map_j . Note that tuple versions in these locations may be different (the later one in map_j).

SURF must therefore identify and reinstate the correct version of each tuple, and so we proceed to the next step.

2. SURF notifies all running applications to abort. This ensures applications bring data to a consistent state from their point of view. Applications use the data locations known to them in performing the abort operations.
3. The system restarts, employing either a *basic* or *enhanced* restart:
 - (a) The *basic* restart mechanism is essentially a fresh start, that is to say:
 - (i) SURF accesses the relevant versions of tuples. Based on their timestamps or other recovery criteria, SURF determines the versions that need re-statement.
 - (ii) These latest versions are used in creating a new complete version of the data in local sites.
 - (iii) The old version of all SURF files is archived.
 - (iv) SURF then restarts afresh from this new complete local data version.
 - (b) The *enhanced* restart mechanism on the other hand makes an effort to identify all locations of valid versions in order to avoid transferring data unnecessarily:
 - (i) SURF examines tuple locations and determines which tuples are already in their new locations (called excluded tuples).
 - (ii) SURF starts a special data move operation that moves all the non-excluded tuples (the excluded tuples are skipped). Once this move operation commences, ordinary applications are notified that the system is operational and is performing a data move.

4.2.4 Partial Data Site Failure

In this scenario, some of the sites on which tuples are stored have failed. This implies that SURF and applications cannot access data on these sites. In this scenario, SURF completely takes these failed sites off the system. An application trying to reach these sites would experience failure. It can abort or re-ask ZK for an alternative location. When a failed site comes up, it will join SURF as a completely new site, and data will be restored to it according to the defined recovery policy. This joining as a new site may be optimized, taking advantage of data already in the site, which is still valid. We do not elaborate on this option; it is similar to the treatment of excluded tuples in the case of a total ZK failure.

4.2.5 Partial Data Site Failures Leading to All Copies of a Tuple Being Lost

In this scenario, all the sites on which a certain tuple t is stored have failed. This leads to the following options (the specific option is a system parameter; it may also involve approval by a human operator):

1. The tuple is inaccessible. SURF monitors the failed sites. Once sufficiently many (a parameter) are up, the proper version of that tuple (according to its recovery policy) is introduced as the tuple version into SURF.
2. An old version of the tuple is brought up from a latest SURF backup (done periodically). This may not be an up-to-date version, but it is a best-effort version.
3. Use the two options above, and resolve the two versions once they are both available (the one from option 2 might have evolved in the meantime).

4.2.6 SURF Has Partially Failed

The SURF architecture has a SURF running process in all operational sites. As long as one SURF process is running, new ones can join the system. If one such process fails during operation, request directed at that process will not be answered. This will identify a failure of that process. A running SURF process may initiate bringing up a SURF process instead of the failed one. Care must be taken that if two such new processes are created, the one with the smaller ID will abort; this is done so that a site will have one SURF process with whom applications communicate.

4.2.7 SURF Has Completely Failed

It is possible that all processes running SURF have failed. In that case, when SURF is restarted in recovery-from-total-failure mode, it first aborts all running ZK instances. It then installs SURF processes on all operational sites. Then, it treats the situation as a total ZK failure (Sect. 4.2.3).

5 Related Work

Zookeeper (ZK) [5] is designed to facilitate coordination in an unreliable distributed computing environment. There are numerous papers and textbooks addressing concurrency control, recovery, and object replication. Early work, dating to the 1980s, may be viewed in [2, 7], and a more up-to-date exposition may be found in [19]. In the distributed computing literature as well, numerous works deal with various notions of consistency, including eventual consistency, e.g., [1, 8, 12, 18]. Our data movement algorithms falls in the intersection of concurrency control algorithms and distributed algorithms (via ZooKeeper). The recovery algorithm is specific to our system and is unlike classical log-based recovery algorithms that are employed today in many systems.

In many papers that deal with data placement in the cloud era, we review a few that are close in nature to SURF's data placement.

1. Application [17] presents an early version of this work.
2. SPANStore [20] optimizes object placement based on price differences while meeting latency and other constraints. Computation is carried out at a *single* provider's data centers, and storage is on various clouds. Object placement is on a per-application basis, and each application has its own SPANStore deployment. There is no proactive object movement as in our work which may lead to delays; an object is moved upon its first use in a new epoch. [13] also considers dynamic migration and replication in view of price differentials, storage classes, and usage; its goal is minimizing monetary cost. They present online deterministic and probabilistic algorithms for predicting behavior.
3. [16] considers a scenario in which an application serves various geographic domains on a *multi-cloud*. Changing (cyclical) workload motivates *virtual machine migration* of both applications and data, in stages. A smart block propagation strategy minimizes latency for moved virtual machines. Migration decisions are optimization-based.
4. [9] considers servers running *components*: front-end, business-logic, or back-end components. A component may be run on more than one server. Components may be storage oriented or computing oriented. The objective is to migrate some components running servers so as to save money and adhere to various enterprise policies (e.g., concerning communication delays).

6 Conclusions

SURF is a technology for data and application placement decisions and actual data movement. Data placement is a complex problem. On the one hand, placement needs to adhere to the enterprise's policy (expressed via business goals) related to the level of security of the data, required quickness of data access, required replication level, data access statistics (frequency), and a myriad of performance parameters (such as capacity and communication characteristics and economic considerations such as budget cap and pricing tables of various providers). Business goals are expressed using the formalism of *goal programs*. The modeling expressed in the resulting goal program (GP) is non-trivial and faithful to actual systems and need be approximated.

In this paper we focus on distributed algorithms that enable data movement with minimal interruptions to applications accessing the data. Using these algorithms, we implemented a SURF platform that allows such applications and tested the platform on a hypothetical application scenario. Extensive long-running simulations reveal the substantial benefits in following SURF's data relocation decisions.

The SURF technology may be embedded in the storage layer of a relational database system (e.g., [4]) or a NoSQL system. This would transform the storage layer into one that continuously optimizes data placement while the data is being accessed by other system layers. We describe an ongoing implementation of such a

system based on HBase [6]. The technology may find ample use in the context of software-defined cloud computing [3].

As the data movement algorithms operate in a complex distributed environment, failures are inevitable. We have built and tested a *recovery* component that eventually brings a failed system back to a legal state.

References

1. H. Attiya, J.L. Welch, Distributed computing—fundamentals, simulations, and advanced topics. Wiley Series on Parallel and Distributed Computing, 2nd edn. (Wiley, London, 2004)
2. A. Bernstein, V. Hadzilacos, N. Goodman, *Concurrency Control and Recovery in Database Systems* (Addison-Wesley Longman, Boston, 1986)
3. R. Buyya, R.N. Calheiros, J. Son, A.V. Dastjerdi, Y. Yoon, Software-defined cloud computing: Architectural elements and open challenges, in *2014 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2014*, Delhi, India, September 24–27, 2014 (IEEE, Piscataway, 2014), pp. 1–12. <https://doi.org/10.1109/ICACCI.2014.6968661>
4. J.C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J.J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, W.C. Hsieh, S. Kanthak, E. Kogan, H. Li, A. Lloyd, S. Melnik, D. Mwaura, D. Nagle, S. Quinlan, R. Rao, L. Rolig, Y. Saito, M. Szymaniak, C. Taylor, R. Wang, D. Woodford, Spanner: Google’s globally distributed database. *ACM Trans. Comput. Syst.* **31**(3), 8:1–8:22 (2013). <https://dl.acm.org/citation.cfm?id=2491245>
5. Foundation, A.S.: Apache zookeeper. <https://zookeeper.apache.org/>. Online Document
6. L. George, *HBase: The Definitive Guide*, 1st edn. (O’Reilly Media, Sebastopol, 2011)
7. J. Gray, A. Reuter, *Transaction Processing: Concepts and Techniques* (Morgan Kaufmann, Los Altos, 1993)
8. S. Gustavsson, S. Andler, Self-stabilization and eventual consistency in replicated real-time databases, in *Proceedings of the First Workshop on Self-Healing Systems, WOSS 2002*, ed. by D. Garlan, J. Kramer, A.L. Wolf Charleston, South Carolina, USA, November 18–19, 2002 (ACM, New York, 2002), pp. 105–107. <https://doi.org/10.1145/582128.582150>
9. M.Y. Hajjat, X. Sun, Y.E. Sung, D.A. Maltz, S.G. Rao, K. Sripanidkulchai, M. Tawarmalani, Cloudward bound: planning for beneficial migration of enterprise applications to the cloud, in *Proceedings of the ACM SIGCOMM 2010 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ed. by S. Kalyanaraman, V.N. Padmanabhan, K.K. Ramakrishnan, R. Shorey, G.M. Voelker, New Delhi, India, August 30–September 3, 2010 (ACM, 2010), pp. 243–254. <http://doi.org/10.1145/1851182.1851212>
10. J. Ignizio, *Goal Programming and Extensions* (Lexington Books, 1976). <https://books.google.co.il/books?id=yEAoAQAAMAAJ>
11. F.P. Junqueira, B.C. Reed, M. Serafini, Zab: high-performance broadcast for primary-backup systems, in *Proceedings of the 2011 IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2011*, Hong Kong, China, June 27–30 2011, (IEEE Compute Society, 2011), pp. 245–256. <https://doi.org/10.1109/DSN.2011.5958223>
12. L. Lamport, Paxos made simple, fast, and byzantine, in *Proceedings of the 6th International Conference on Principles of Distributed Systems. OPODIS 2002*, ed. by A. Bui, H. Fouchal, Reims, France, December 11–13, 2002. *Studia Informatica Universalis*, vol. 3 (Suger, Saint-Denis, rue Catulienne, 2002), pp. 7–9
13. Y. Mansouri, A.N. Toosi, R. Buyya, Cost optimization for dynamic replication and migration of data in cloud data centers. *IEEE Trans. Cloud Comput.* **7**(3), 705–718 (2019). <https://doi.org/10.1109/TCC.2017.2659728>
14. C. Romero, *Handbook of Critical Issues in Goal Programming* (Pergamon Press, 1991)
15. A.W. Services, Amazon s3. <https://aws.amazon.com/s3/>. Online Document
16. Z. Shen, Q. Jia, G. Sela, B. Rainero, W. Song, R. van Renesse, H. Weatherspoon, Follow the sun through the clouds: Application migration for geographically shifting workloads, in

- Proceedings of the Seventh ACM Symposium on Cloud Computing*, ed. by M.K. Aguilera, B. Cooper, Y. Diao, Santa Clara, CA, USA, October 5–7, 2016 (ACM, New York, 2016), pp. 141–154. <http://doi.acm.org/10.1145/2987550.2987561>
17. O. Shmueli, I. Shaked, Moving replicated data in a cloud environment (2018). <https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2018185771>. Application No.: PCT/IL2018/050404
 18. R. Strickland, *Cassandra High Availability* (Packt Publishing, 2014)
 19. G. Weikum, G. Vossen, *Transactional Information Systems: Theory, Algorithms, and the Practice of Concurrency Control and Recovery* (Morgan Kaufmann, Los Altos, 2002)
 20. Z. Wu, M. Butkiewicz, D. Perkins, E. Katz-Bassett, H.V. Madhyastha, *SPANStore*: cost-effective geo-replicated storage spanning multiple cloud services, in *ACM SIGOPS 24th Symposium on Operating Systems Principles, SOSP '13*, ed. by M. Kaminsky, M. Dahlin, Farmington, PA, USA, November 3–6, 2013 (ACM, New York, 2013), pp. 292–308. <https://doi.org/10.1145/2517349.2522730>

The Abaco Platform: A Performance and Scalability Study on the Jetstream Cloud



Christian R. Garcia, Joe Stubbs, Julia Looney, Anagha Jamthe,
Mike Packard, and Kreshel Nguyen

1 Introduction

Abaco (Actor-BASED COntainers) [1] is a hosted, functions-as-a-service application programming interface (API) which combines Linux container technology with the Actor Model of Concurrent Computation [2]. Also referred to as a “serverless” platform, Abaco enables the rapid development of reactive, event-driven architectures requiring minimal long-term maintenance. Funded since 2017 by the National Science Foundation, Abaco serves the national research community with the ability to run high-throughput, low-latency workloads concurrently for faster execution. In its first 2 years in production, Abaco has been adopted by several projects and has supported the invocation of over 70,000 functions collectively running for more than 20 million seconds.

Users register new actors in Abaco by making an API request that includes a reference to a publicly available Docker image to use for the actor; in response, Abaco generates and returns a URI associated with the new actor. Users can then use the URI to send messages to the actor. For each message sent to an actor, Abaco puts the message on an internal queue assigned to the actor, referred to as the actor’s “mailbox” or “inbox.” For each queued execution, as Abaco’s internal compute resources become available, the system launches a container from the actor’s image

C. R. Garcia (✉) · J. Stubbs · J. Looney · A. Jamthe · M. Packard
Texas Advanced Computing Center, University of Texas at Austin, Austin, TX, USA
e-mail: cgarcia@tacc.utexas.edu; jstubbs@tacc.utexas.edu; jlooney@tacc.utexas.edu;
ajamthe@tacc.utexas.edu; mpackard@tacc.utexas.edu

K. Nguyen
Computational Engineering Dept of Aerospace Engineering, University of Texas, Austin,
TX, USA
e-mail: kreshel@utexas.edu

and injects the original message data into the container. Most Abaco executions start within 1 or 2 seconds of a message being sent; however, in some cases, users will queue tens of thousands of messages in a short period of time for a single actor, in which case, the executions will run over several hours or days.

The resource requirements of workloads on Abaco can vary greatly depending on the nature of the computation being performed. Abaco provides users with different ways of interacting with the API to facilitate executions. In particular, users have the option of manually scaling the resources assigned to an actor or using Abaco's autoscaling feature to dynamically scale resources based on the actor's mailbox size. In this paper, we set out to experimentally determine the differences in performance between manual scaling and autoscaling under two different workloads and two different metrics: "FLOPS," which determines the amount of work that can be achieved by a system at a given time, and "hashrate," which is a measure of performance that became popular with the emergence of Bitcoin [3] technology.

1.1 Abaco Background

The Abaco system uses internal agents referred to as "workers" to facilitate the processing of actor messages. When a worker is created, it is assigned to exactly one actor, and it subscribes to the internal Abaco queue corresponding to the actor's mailbox, defined in RabbitMQ [4]. In response to receiving a message intended for its assigned actor, a worker starts an actor container from the defined image associated with the actor and injects the message data into the container, either in an environment variable in the case of text data or a UNIX domain socket in the case of binary data. The worker then supervises the actor execution, monitors for the actor process to exit, and collects resource usage and log data along the way. A given worker does not retrieve a new message from the actor's queue until the current execution is finalized. It follows that, for a given actor, the number of messages being processed concurrently at any given time is no more than the number of workers assigned to the actor. In particular, an actor with one worker only processes one message at a time.

In many applications, it is critical that messages are processed sequentially and Abaco formalizes this notion through a property called "stateless" provided at actor registration. If an actor is registered with the "stateless" property set to false, Abaco will never start more than one worker for the actor. This feature distinguishes Abaco from many other functions-as-a-service offerings and is an important aspect of the Actor Model more generally.

However, for actors registered with the stateless property set to true, Abaco can increase or decrease the number of workers associated with an actor. Abaco provides an endpoint in its HTTP API that users can use to create additional workers or shutdown existing workers for actors they have access to. This approach is referred to as "manual scaling."

1.2 Autoscaling

The Abaco autoscaler will automatically scale the number of workers associated with an actor according to the size of the actor's mailbox. The autoscaling will scale up when an actor has at least one message in its inbox. Similarly, when the actor has zero messages in its inbox, the autoscaler will shut down workers for that actor that are not currently overseeing a running execution. There is also a limit to the number of workers an actor can have. If this limit is reached, then Abaco will stop the scale-up process until the number of workers for that actor has reduced below the limit.

The autoscaling in Abaco has been developed on top of Prometheus, an open-source time series database and monitoring server [5]. Prometheus functions by scraping plain-text values that it converts into time series data. Prometheus can be configured to repeatedly scrape values from specific APIs on a given time interval. The algorithm for autoscaling is as follows. The Abaco metrics API endpoint produces plain-text values of the current actor inbox sizes, as well as how many workers are assigned to each actor. Prometheus scrapes the instantaneous mailbox sizes using the Abaco metrics endpoint every 5 seconds. The autoscaler then uses this data to determine whether it should scale up, scale down, or neither. This allows Abaco to provide more resources to those actors which require it while removing resources once they are no longer needed. If an actor has at least one message, but it has reached its limit of workers, and then neither scale-up nor scale-down will occur.

2 Experiment Design

Harnessing the full potential of high-performance computing at a user level has become increasingly difficult as the field has grown in scale and complexity. The Abaco platform attempts to alleviate this issue by giving users an additional tool in HPC. This study seeks to answer the research questions below by analyzing various performance metrics in order to better understand the use, potential, and limitations of the Abaco platform.

2.1 Research Questions

- What is the difference in worker creation rate between a manually scaled worker pool and using the autoscaler?
- What is the performance of Abaco at different node sizes, and how does it compare to the theoretical limit of the hardware utilized?

- What are the scaling limits of the Abaco platform, and what are the causes to those limits?

2.2 Experiment Overview

Our experiment compares the performance of the Abaco autoscaler versus manually scaling actor's worker pools. In order to gain solid insights and apply them to our research questions, we must compartmentalize our testing. First we must measure strictly the overhead due to manual scaling versus autoscaling when it comes to worker creation rates, now to be called the "scaling test." Secondly we must measure the total performance difference between manual scaling and autoscaling with some form of performance metric, now to be called the "comprehensive test."

2.3 Scaling Test Overview

For our scaling test, we must test the difference in worker creation rate for manual scaling and autoscaling. To clarify, manual scaling consists of determining the exact amount of workers wanted and initializing workers up until that point with an API call. This prompts Abaco to immediately begin creating workers to a set point. Autoscaling would instead intelligently create workers based on the amount of messages in an actor's queue, efficiently managing compute resources. Unlike manual scaling, autoscaling would then require some bypass in order to measure purely the rate of worker creation without running actors by sending messages. In order to do this, we will manually set a variable in code so that the autoscaling logic always needs to create more workers. This allows us to compare the worker creation rate between the autoscaler's logic and manual scaling with ease.

2.4 Comprehensive Test Overview

For our comprehensive test, we must test the difference in performance from start to end. In order to do that, the test must consist of three main tasks. First we must create an actor, second we must create workers for that actor, and third we must run the actual computations for that actor. Actor creation is identical no matter whether the autoscaler capability is used or not. Worker creation varies and is measured independently in the scaling test, but it is additionally measured here to gain an overall picture of performance. The actual computations are also dependent variables which can be measured using two sets of performance metrics, and they allow us to measure any differences due to workload or work type. Additionally,

we measure how different hardware allocations affect the results by making use of Jetstream to run tests on different node sizes.

2.5 Performance Metrics

In this section we describe the performance metrics utilized in the comprehensive test, FLOPS, and hashrate, along with the theoretical bounds used in test setup.

2.5.1 FLOPS

The four basic mathematical floating-point operations performed by computers are addition, subtraction, multiplication, and division. The amount of these operations performed on a per-second basis is referred to as floating-point operations per second, abbreviated as FLOPS. FLOPS of a computer system acts as a metric for the amount of work a computer can perform in a given time. In this experiment we use the amount of FLOPS to compare different testing scenarios and gauge system overheads, slowdowns, and most importantly, scalability.

There are two parts to the FLOPS evaluation experiment. First, a large-scale distribution of relatively quick work is set up to test the autoscaling performance of the platform, “quick FLOPS.” Second, a smaller-scale distribution of relatively slow work is set up to achieve peak FLOPS by removing overhead in worker deployment, “slow FLOPS.”

The “work” mentioned above is a Python script that when executed takes three inputs: the amount of threads the script should utilize, standard deviation for randomizing a square matrix, and the size of the two square matrices. The script then calculates the dot product of these two random square matrices as the “work of the experiment.” This script is packaged as a Docker Hub image at `abacosamples/abaco_perf_flops` [6].

To calculate the amount of FLOP in each execution, we use Eq. (1), where S is the input size, described above. This equation calculates the amount of FLOP in a dot product operation between two square matrices [7]:

$$\text{FLOP} = (S + S - 1) * S^2 \quad (1)$$

From there, in order to calculate the speed of one whole trial, we use Eq. (2) to calculate FLOPS. We do this by taking the amount of FLOP per executions; multiplying it by the amount of executions, E , of each trial; and dividing by the amount of elapsed time, t , from trial start to end:

$$\text{FLOPS}_{\text{real}} = \frac{(2S^3 - S^2) * E}{t} \quad (2)$$

2.5.2 Hashrate

The second performance metric used in our experiment is hashrate. Hashrate is the number of hashes per second, where a hash is one iteration of a SHA256 hashing function, which creates a hashed block [8]. This measure of performance became popular with the rise of Bitcoin [3], where mining a block with certain parameters gave a reward in the form of bitcoins. Without using an identical hashing function, language, and algorithm to Bitcoin, which is proprietary, or some other proof-of-work function, we are unable to compare our hashrate with the results from other computers. Instead we will hash a set amount of blocks and compare results gathered through the testing.

To actually calculate hashrate of the system, we register an actor using the `abacosamples/abaco_perf_hashrate` [9] image, available on Docker Hub. This image contains a script that runs Python's `hashlib.sha256()` function a given number of times. Each iteration of this `hashlib.sha256()` function is a hash of a block. Our experiment runs each execution running until three million hashes are calculated. Hashrate is given in Eq. (3), where Hr is the hashrate, h is the hashes, and t is the elapsed time from trial start to end:

$$\text{Hr} = \frac{h}{t} \quad (3)$$

2.6 Obtaining Theoretical Bounds

While we have an established method of obtaining FLOPS and hashrate, we're still in need for a result to compare our data against. With an addition of a comparative result, we gain access to additional insights on system overhead and the real limits of our testing.

We use Eq. (4) to obtain the theoretical limit of FLOPS for a single server [10]. In this equation N is the number of cores per CPU, F is the average frequency of these cores, and O is the operations per cycle for each CPU:

$$\text{FLOPS}_{\text{theory}} = N_{\text{cores}} * F_{\text{avg}} * O_{\text{cycle}} \quad (4)$$

Plugging into the above equation knowing that the Jetstream server cluster is using 6 cores of a v3 Intel Xeon E5-2680 turbo-boosted to 3.30 GHz, and running 16 operations per cycle gives us a theoretical max speed of 316.8 GFLOPS per node. This number now acts as the theoretical fastest speed that our servers could ever possibly achieve.

Along with a theoretical limit for FLOPS, we also have a practical limit, which is the speed of our experiment script running solely on a Jetstream compute node. This is the practical limit for our experiment, and any change from that number would be caused by overhead from Abaco, Docker, Python, networking, etc. Due to the fact

that this is what is practically possible, it will more than likely be the most important metric to compare against as it points out more clearly the difference between one possible result and another.

Hashrate on the other hand is determined empirically and thus does not have a theoretical max that we can calculate. Instead, similar to FLOPS, we run our hashrate script solely on a Jetstream compute node, giving us our so called “practical limit.” which allows us to compare and view the effects of system overhead.

3 Experimental Setup

We divide the experiment setup into two parts: deployment and validation. We first deploy the servers that run Abaco, its components, and the compute nodes that Abaco will utilize to spawn new workers on. Secondly, we run the test suite to conduct the performance studies. Our entire testing repository, along with Docker images, is hosted on GitHub, [TACC/abaco-autoscaling](https://github.com/TACC/abaco-autoscaling), with READMEs detailing the exact instructions to reproduce the experiment. In this section we describe an overview of the experimental process so that users can implement this method in their workflows.

3.1 Resource Configuration

We begin with configuring the resources needed for this experiment. All servers are hosted by the NSF’s scalable cloud system for XSEDE, Jetstream [11]. This service allows for the creation and configuration of virtual machines (VMs), or “nodes,” which gives our experiment the ability to scale. Jetstream uses OpenStack [12] for resource management and gives us the ability to deploy servers through the command line interface (CLI). Although this paper will continue to reference the resources used, it’s important to note that the Abaco platform is capable of running on any cluster of Linux nodes.

All nodes in this experiment have the following specifications: CentOS Linux version 7.6.1810, kernel version 3.10.0-957.5.1.el7.x86_64, Docker version 18.09.5, and Docker Compose version 1.24.0. All Abaco nodes in this experiment are Jetstream m1.quad nodes which have 10 GB RAM, 20 GB SSD storage, and 4 vCPUs. A vCPU in this case is one core of a Intel Xeon E5-2680 v3, which can turbo-boost from 2.50 GHz to 3.30 GHz.

3.2 *Resource Deployment*

In this section we describe the automated deployment process used by the test program to create the cluster of nodes and initialize the Abaco platform exercised by the test trial. At a high level, deployment consists of (1) creating five OpenStack instances for each of the dedicated Abaco components (MongoDB, Redis, RabbitMQ, Prometheus, and the Abaco web services), (2) creating a number of OpenStack instances corresponding to the cluster size of the trial (these are the Abaco “compute nodes”), and (3) pulling and initializing the services.

All scripts used by the test program for automating the deployment are maintained in the GitHub repository for this project [13], within the `/deployment` folder. The `README.md` file, included in that directory, provides an in-depth description of all of the scripts available for the users.

To simplify the process of running multiple trials, the performance test suite’s git repository was designed to be cloned to a single persistent instance running within the OpenStack network where the node clusters for the different trials will be provisioned. Scripts that make use of the OpenStack CLI to start or stop instances require OpenStack authentication. The root directory of the repository contains a shell script, `openrc-script`, which prompts the user for a password when ran using command `./openrc-script`. This script sets environment variables that will authenticate a user with OpenStack. It’s worth noting that an entire Abaco installation can run correctly on just one node, but for the experiment, we chose to separate Abaco’s components across five separate nodes for resource distribution and debugging purposes.

Additionally, the test scripts make use of a few predefined OpenStack instance images. Using prebuilt images that included base software such as the Docker container runtime, Docker-compose, etc., we were able to significantly reduce the overall runtime of the test suite across multiple trials, as the instances were burned down and recreated between runs. The images used by the tests are available on the Jetstream system and can be made available to import to other OpenStack clusters upon request.

The Abaco system is not ready for use until the compute nodes are created and ready for use however. Creation and management of the compute nodes are also automated with scripts in the `/deployment` folder. For example, the `up_instances` script creates a specified input number of OpenStack `m1.medium` compute nodes using the prebuilt `perf-abaco-compute` OpenStack image. The `up_abaco` script starts the Abaco services (packaged as Docker containers) on the various instances. The `up_abaco` script is a convenience wrapper around an Ansible playbook designed to interact with sets of instances running on an OpenStack cluster, also included in the repository.

3.3 Validation Setup

The test suite itself is also automated and requires little human intervention once the Abaco system against which it runs on has been instantiated. The `/test_suite/tests` folder within the GitHub repository includes all the Python tests for this experiment. The master test script, named `run_tests.sh`, runs the tests for all specified node sizes, each time deleting nodes to reach the specified amount of nodes and re-initializing all containers.

To run the comprehensive tests with the autoscaler turned on, the `run_tests.sh` script must be modified by uncommenting the Python scripts that begin with “scaling,” and the instances must be redeployed with an updated version of the `abaco.conf` file, included in the repository. It’s also worth noting that the Prometheus component is only needed when running with the autoscaler turned on. Complete details are provided in the project repository on GitHub.

3.4 Validation

The execution of our experiment suite has both the scripts for our scaling test and our comprehensive tests.

Our scaling tests consist of 5 trials at 13 different node sizes. The test is setup so that there is one actor on every node, so at a node size of 80 there are 80 actors. Once the test begins, starting immediately, the only goal of Abaco is to create workers. In the case of manual scaling, a manual API call requests 5000 workers per actor. In the case of autoscaling, the autoscaling logic of Abaco is manually reconfigured so that the autoscaling logic always “thinks” that there are 5000 messages in every actor’s message queue; thus 5000 workers want to be created. We let this test continue on for 10 min before shutting down Abaco and collecting our results. In order to trick the autoscaler in Abaco, we manually fix the `current_message_count` variable to 5000 in Abaco’s `actors/controllers.py` file.

Our comprehensive test on the other hand has five trials of six jobs—three using manual scaling and three using autoscaler—at ten different node sizes. The first of the three jobs with manual scaling is a quick FLOPS test. The test is setup with six Abaco actors per node to have one actor per node. Each of these actors are given five executions to complete, and each of these executions consists of doing the dot product of two square matrices of dimensions 8000 by 8000. The second test using manual scaling is the hashrate test and is set up similarly to quick work FLOPS except for the fact that each actor is given six executions and each execution is meant to complete 3,000,000 SHA256 hashes. The third test using manual scaling is the slow FLOPS test. This test has one Abaco actor per node so that each actor gets six cores to run on. Each of these actors is given five executions to complete, and each of these executions consists of doing the dot product of two square matrices

of dimensions 25,000 by 25,000. The last three tests are exactly the same as the first three, but they utilize the Abaco autoscaler for worker management.

For this paper, we use the Abaco system configurations described in Table 1, for each test type. These parameters are set in the `abaco.conf` file found in the `deployment/abaco_files` folder of the repository. These parameters act as the upper bounds of available workers in each test. For the scaling tests, we leave everything uncapped to allow as many workers as possible to be created so that we can get the best worker creation rate achievable. In the comprehensive test, the bounds resemble configurations in TACC’s existing production deployment of Abaco and ensure that we don’t scale up our workers and cause bottlenecks due to inefficient process distribution. For example, scaling to a number of workers greater than the amount of cores available on the node could result in CPU priority issues resulting in slight slowdowns due to multiple processes attempting to run on one core at once.

4 Findings

We conduct experiments to answer the research questions stated in Sect. 2.1. From our scaling tests, we gather data on the rate of worker creation at different node sizes for resulting from manually scaling and autoscaling. Our comprehensive tests on the other hand do the same, except they additionally measure our two performance metrics in order to visualize any fall-off in performance.

Our experimental findings broadly follow the patterns that we expect in terms of worker creation rate and performance drop-off. Using these insights we propose to answer the research questions posed by this paper in the following sections.

4.1 *Difference in Worker Scaling Rate Between Scaling Types*

By independently gathering data on worker creation rate in the form of our scaling tests, we can clearly see the differences during worker creation due to autoscaling.

Table 1 Worker configurations per test type

Test type	Max workers per node	Max workers per actor
Scaling test	5000	5000
Quick FLOPS	6	30
Slow FLOPS	1	1
Hashrate	6	36

Figures 1 and 2 both are split into two subplots for easier viewing and analysis. Experiments on node sizes 1 through 40 are on the left plot of each figure and node sizes 50 through 100 are on the right plot of each figure.

In Fig. 1 we see two important elements. First, from 1 to 40 nodes, the manually scaled worker creation rate increases linearly. Secondly from 50 to 100 nodes, there seems to be a worker creation slowdown at around 200 seconds, when around 8000 workers are created. While this particular set of circumstances is unlikely to repeat itself in a real workflow, we can see that the rate of worker creation reaches some upper limit that results in a slowdown of worker creation.

In Fig. 2 we can see that from 1 to 40 nodes worker creation rate increases in a relatively regular fashion and reaches a ceiling from 50 to 100 nodes. This can likely be attributed to the way Abaco queues up and creates workers once they are requested. While there may be resources to create more workers, the autoscaler requests those resources at set time increments and caps off the worker creation rate.

Figure 3 shows us the overall average worker creation rate for both manual and autoscaler cases and compares them. Here we can see that the worker creation rate slows down in the manually scaled case at around a node size of 70 when overall average rate begins to drop off. We can also see the rate gradually increasing in the autoscaling case as worker creation rate rises with additional nodes and then reaches a plateau.

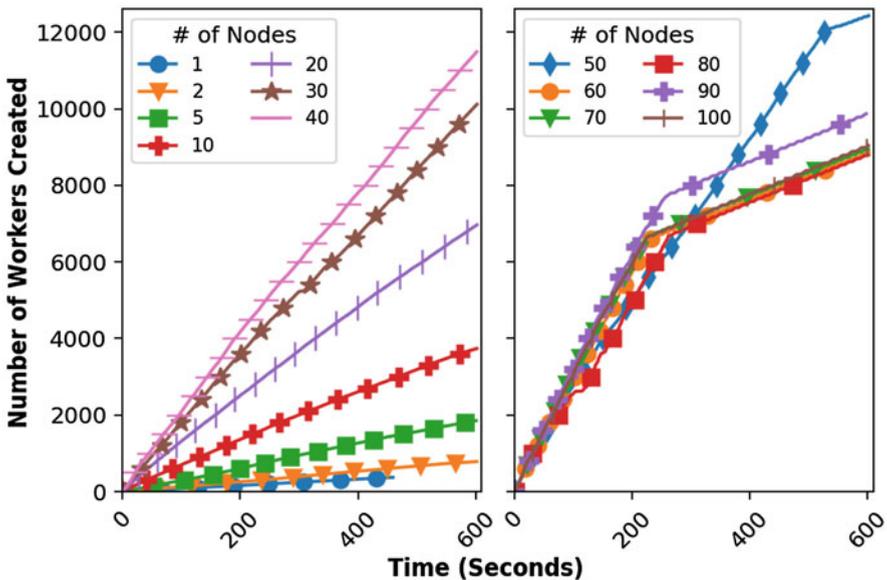


Fig. 1 Rate of worker creation in the manually scaled case

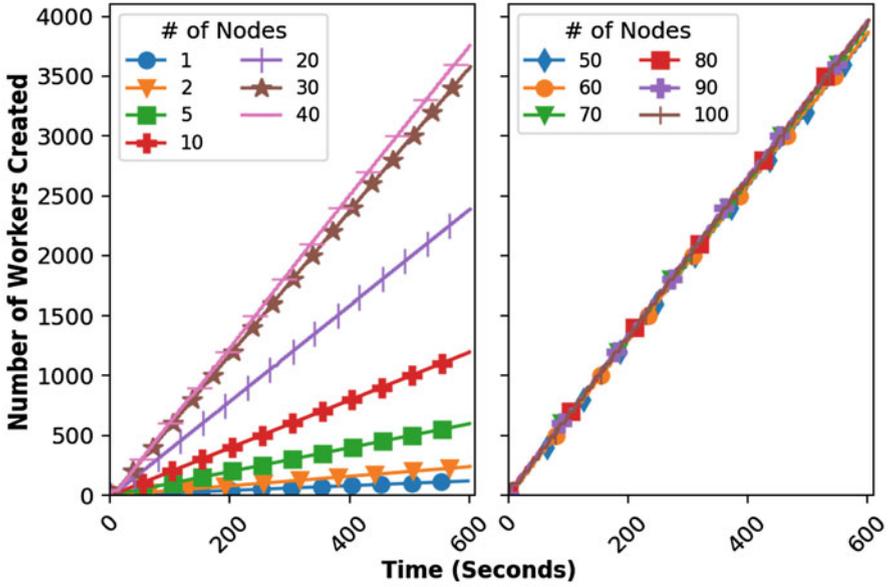


Fig. 2 Rate of worker creation in the autoscaler case

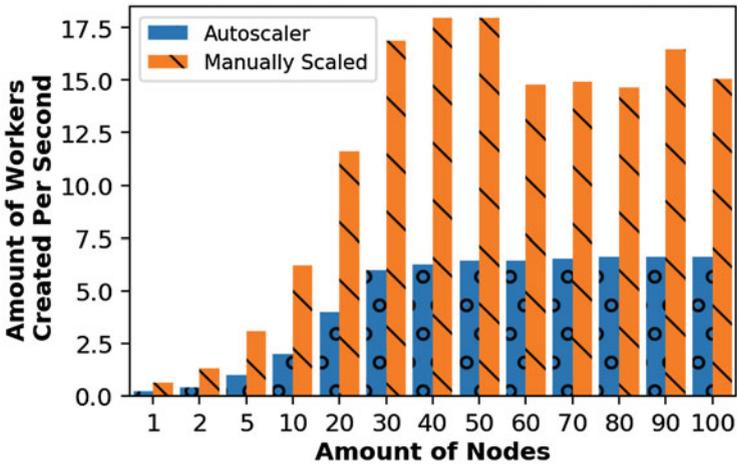


Fig. 3 Comparison of worker creation rates

The data gathered gives useful information to users about the Abaco platform. We see that the manually scaled case has a higher rate of worker creation. The necessity of this performance is somewhat negligible however due to a use case of this variety being particular rare. The autoscaler rate has a ceiling of about 6.5 workers created every second. Over the course of 1 min, any node size greater than around 30 would produce around 390 workers. Ten minutes would produce around 3900 workers. This amount of workers per node would create a bottleneck on resources solely due to having so many Docker containers running in parallel. Even more resource hungry would be manually scaling workers at a rate of 15 workers per node which is possible at a node size above 30. While possible however, this would produce a prohibitively large amount of workers on a single node.

Overall, we see that Abaco performs exceptionally well during worker creation. Assuming a user is running demanding tests, worker creation time would be a small fraction of total run time and will far exceed the performance necessary for most users. We draw this conclusion from the fact that a typical VM or environment to run work on would take a few seconds to initialize. In the case of Abaco, with the autoscaler on, it would only take 4 seconds to have 26 workers ready and executing, an amount that many people would find difficult to utilize. In the case of manually scaled testing, we do run into performance issues; it is important to note though that these issues don't arise until around 7000 workers are created and is likely due to the rate of worker creation being too fast.

4.2 Analysis of Performance at Different Node Sizes

One of the most important questions to ask when using Abaco is how much performance is a user ready to trade in for the ease of use and accessibility of the Abaco autoscaler. Due to this question, we crafted our comprehensive tests in order to analyze the amount of performance spent for convenience.

Column 1 of Table 2 is the ratio of autoscaler performance to manually scaled performance. From the results we can see that the overall trade-off when an execution is running is nearly indistinguishable with the autoscaler even being marginally better than the manually scaled performance in the quick work tests.

Table 2 Performance ratios at a node size of 89

	Ratio of autoscaler speed to manually scaled speed	Ratio of manually scaled speed to theoretical speed	Ratio of Jetstream speed to theoretical speed	Ratio of manually scaled speed to Jetstream speed
Fast FLOPS test	102.4%	65.1%	71.5%	80.2%
Slow FLOPS test	99.1%	67.7%	81.2%	94.7%
Hashrate test	99.5%	N/A	N/A	92.5%

This statistic follows our expected patterns as the autoscaler and manually scaled tests should only differ in how workers are created, not performance in executions.

In the second column of Table 2, we see that the quick work tests are 65.1% of theoretical performance, the slow work tests are 67.7% of theoretical performance, and the hashrate tests does not have a theoretical performance due to the empirical nature of hashrate testing. The third column of Table 2 gleans perspective on these numbers by comparing the theoretical hardware performance to our Jetstream performance. In one case, Jetstream is 71.5% of the theoretical performance, while in the other, it is 81.2% of the theoretical performance. In essence the theoretical hardware performance is an unreasonable metric to compare against when our testing hardware could only practically achieve around 75% of that performance. Thus the important metric to compare to is the practical result, which is Jetstream performance.

In the fourth column of Table 2, we see the manually scaled speed as a percentage of Jetstream speed. In the quick work tests, we see 80.2% of Jetstream speed. This is due to a combination of Docker overhead, Abaco overhead, and overhead due to the node. For instance, running a multitude of quick tests means that the node's CPU is constantly going in and out of working and must constantly change clock speed from resting to turbo. In the other two tests, performance is 92.5% and 94.7% of the Jetstream speed, which again can be attributed to the same causes.

Figure 4 visualizes the results of Table 2. With this figure we see that performance increases linearly as more nodes are added to the tests. and there's no any irregularities when executions are being run. This gives us a sense of scale in regard to the amount of work that the Abaco platform is capable of running. At 89 nodes, the slow FLOPS test reaches 19 TFLOPS of performance. This is equivalent to the theoretical performance of 13 Stampede2 [14] nodes.

To answer the research question, overall the performance trade-off of using the Abaco platform is minor as compared to the Jetstream performance, and the trade-off of using the Abaco autoscaler is non-existent once executions are actually being run.

4.3 *The Scaling Limits of Abaco*

A question that must be asked for a researcher is: how far can the Abaco platform scale out before issues arise and potentially affect usage? In our experimentation we ran into several hurdles; many were easily fixed, while others will need future improvements made to the Abaco system to ensure peak performance no matter the case.

The first issue that we faced came from Docker Hub. When testing the performance of the platform, if too many workers were requested at once, Abaco would eventually begin receiving HTTP status codes 429, "Too Many Requests." This would result in a platform error that attempted to scale back the number of workers to clear platform congestion. To alleviate this issue going forward, Abaco should

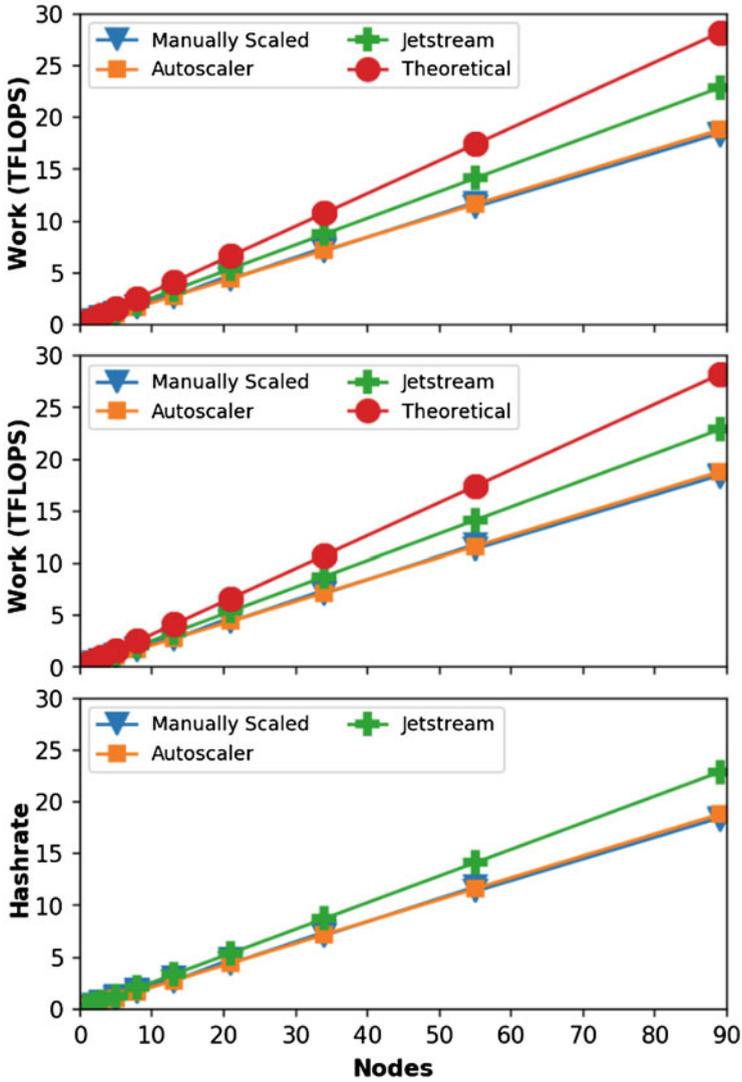


Fig. 4 TFLOPS of different tests based on node size Top: Fast FLOPS tests Center: Slow FLOPS tests Bottom: Hashrate tests

either set a capped Docker request rate or add an actor flag that bypasses pulling worker images unless they are missing.

Two more issues seen were related to node resources. The first was RabbitMQ restricting any incoming messages when a node's RAM was sufficiently in use. This was alleviated by setting the `RABBITMQ_VM_MEMORY_HIGH_WATERMARK` environment variable to a higher percentage. The second resource issue came from MongoDB, also consuming a large amount of node RAM. This issue resulted in node slowdown and eventual slow responsiveness and was most likely the cause of worker creation slowdowns in the case of very large node sizes.

Another major issue that was observed involved Abaco's use of its Redis database. Abaco stores all runtime information for a given actor under a single key in the Redis database. Initial versions of the experiment sent all messages in a given test to a single actor. At very high node counts, performance suffered as a result of Redis optimistic locking when information was being written to the actor record. This was alleviated by changing the tests to use multiple actors (e.g., one actor per node) and dividing the work evenly across the actors in the experiment. Abaco is nearing completion of a set of changes to its database usage to improve query and write performance in the rare cases that one actor doing so much work could be more beneficial.

Overall, the primary symptoms of scaling issues did not come in any fatal outcomes but in slowdowns of the Abaco systems.

5 Related Work

Abaco draws comparison to and inspiration from a number of existing software systems.

5.1 *Functions-as-a-Service*

There are commercial offerings from AWS Lambda [15], Google Cloud Functions [16], Microsoft Azure Functions [17], and IBM Apache OpenWhisk Functions [18]. However, many of these offerings have limits on memory allocation and runtime duration. Some also have limitations on which languages can be used. AWS Lambda allows many different languages but limits users to 1536 MB of memory allocation and 300 seconds of runtime, while Google Cloud Functions only allows Java, Node.js, Python, and Go and has a maximum duration of 540 seconds, and Microsoft Azure functions has no limit on execution time limit but limits the amount of functions running at once to 10.

The closest in spirit to Abaco would be the open-source project OpenFAAS [19], which provides functions-as-a-service based on Docker images. Unlike Abaco,

however, it requires the function container to run an HTTP server. It also does not include the Actor Model and several other features that are part of Abaco.

5.2 *Containers-as-a-Service*

Commercial examples of containers-as-a-service include Amazon’s Elastic Container Service (ECS) [20] and Google’s Container Engine [21]. Although these services allow the use of arbitrary container images, they lack the actor-based architecture that is part of Abaco’s design, making them better suited for long-running server daemons.

5.3 *Distributed Computing Platforms*

Platforms such as Apache Spark [22], Apache Storm [23], iPython parallels [24], and AWS Kinesis [25] provide features similar to Abaco’s scientific functions. These systems even support additional paradigms such as inter-process communication (IPC) and provide better performance. For Abaco, scientific functions only ever attempt to achieve pleasantly parallel compute jobs, and its goal is to make them more accessible.

6 Conclusion

In this study, we tested the actual performance of Abaco to compare it with the theoretical bounds of the system. We measured the differences in performance and worker creation rate between manually scaled workers and the autoscaler and evaluated the scaling limits of Abaco and their causes. From our findings we conclude that the Abaco platform greatly automatizes additional setup and frees up valuable user time at little to no extra cost compared to a manually scaled setup. If we were to improve upon this experiment, we would want to overhaul Abaco in order to optimize scaling to more compute nodes. Before this experiment the scalability of Abaco was unknown. However, with the fixes mentioned before and hardening of some systems, Abaco has the potential to grow to an even greater extent and give researchers an even better tool to make use of HPC resources. Overall the study demonstrates the practicality of using Abaco to simplify workflow and using the Abaco autoscaler to further reduce user time needed.

Acknowledgments This material is based upon work supported by the National Science Foundation Office of Advanced CyberInfrastructure, award number 1740288. This work used the

Extreme Science and Engineering Discovery Environment (XSEDE) Jetstream resource at the TACC through allocation CCR190017.

References

1. J. Stubbs et al., Rapid development of scalable, distributed computation with Abaco, in *10th International Workshop on Science Gateways* (Science Gateways Community Institute, 2018)
2. G. Agha, *Actors: A Model of Concurrent Computation in Distributed Systems* (MIT Press, Cambridge, 1986)
3. S. Nakamoto, Bitcoin: a peer-to-peer electronic cash system (2009). <http://www.bitcoin.org/bitcoin.pdf>
4. A. Wood, *Rabbit MQ: For Starters* (CreateSpace Independent Publishing Platform, North Charleston, 2016)
5. Prometheus, Prometheus. <https://github.com/prometheus/prometheus>
6. Docker hub flops image (2020). Accessed 17 Feb 2020. https://hub.docker.com/repository/docker/abacosamples/abaco_perf_flops
7. Numpy dot product (2020). Accessed 17 Feb 2020. https://www.tutorialspoint.com/numpy/numpy_dot.htm
8. What is hashrate? (2020) Accessed 17 Feb 2020. <https://www.buybitcoinworldwide.com/mining/hash-rate/>
9. Docker hub hashrate image (2020). Accessed 17 Feb 2020. https://hub.docker.com/repository/docker/abacosamples/abaco_perf_hashrate
10. D.M.R. Fernandez, Nodes, sockets, cores and flops, oh, my (2020). Accessed 17 Feb 2020. <https://www.shorturl.at/kUZ12>
11. C.A. Stewart, T.M. Cockerill, I. Foster, D. Hancock, N. Merchant, E. Skidmore, D. Stanzione, J. Taylor, S. Tuecke, G. Turner, M. Vaughn, N.I. Gaffney, Jetstream: a self-provisioned, scalable science and engineering cloud environment, in *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*, ser. XSEDE '15 (ACM, New York, 2015), pp. 29:1–29:8. <http://doi.acm.org/10.1145/2792745.2792774>
12. A. Shrivastwa, S. Sarat, K. Jackson, C. Bunch, E. Sigler, T. Campbell, *OpenStack: Building a Cloud Environment* (Packt Publishing, 2016)
13. Github—abaco autoscaling (2020). Accessed 17 Feb 2020. <https://github.com/tacc/abaco-autoscaling>
14. Stampede2 (2020). Accessed 17 Feb 2020. <https://www.tacc.utexas.edu/systems/stampede2>
15. Aws Lambda (2020). Accessed 17 Feb 2020. <https://aws.amazon.com/lambda/>
16. Google Cloud Foundation (2020). Accessed 17 Feb 2020. <https://cloud.google.com/foundation-toolkit/>
17. Microsoft azure (2020). Accessed 17 Feb 2020. <https://azure.microsoft.com/>
18. Apache openwhisk (2020). Accessed 17 Feb 2020. <https://openwhisk.apache.org>
19. Openfaas (2020). Accessed 17 Feb 2020. <https://www.openfaas.com>
20. Amazon elastic container service (2020). Accessed 17 Feb 2020. <https://aws.amazon.com/ecs/>
21. What is gke? (2020). Accessed 17 Feb 2020. <https://www.aquasec.com/wiki/display/containers/Google+Container+Engine>
22. Apache spark (2020). Accessed 17 Feb 2020. <https://spark.apache.org>
23. Apache storm (2019). Accessed 17 Feb 2020. <http://storm.apache.org>
24. Ipython parallel (2020). Accessed 17 Feb 2020. https://ipython.org/ipython-doc/stable/parallel/parallel_intro.html
25. Amazon kinesis (2020). Accessed 17 Feb 2020. <https://aws.amazon.com/kinesis/>

Enterprise Backend as a Service (EBaaS)



Gokay Saldamli, Aditya Doshatti, Darshil Kapadia, Devashish Nyati, Maulin Bodiwala, and Levent Ertaul

1 Introduction

Web application is nothing but a piece of software running on a remote computer which can then be accessed by anyone and at anytime over the internet. The idea of not having the code on our local computer and still be able to access it has played a major role in people accepting the trend of web applications. But this development of web applications would not have been this popular if the concept of RESTful (REpresentational State Transfer) system had not come into the picture. REST system was introduced as a protocol for exchanging data over the internet and by doing so revolutionized the development of web services. As discussed above, web application is majorly comprised of front-end and back-end. Frontend deals with how the data should be shown to the user and back-end deals with what data need to be shown. In the past when REST system was not introduced, people were controlling the data on the front-end side and in turn making the whole process heavy. Since the introduction of REST and RESTful APIs, developers have started to design the functionalities as APIs which will then be consumed on the front-end. This separate piece of RESTful code is a major part of the back-end. In today's world, one can see no projects that does not use the RESTful system for the creation of professional web services. Major companies like Yahoo, Facebook, Google, and many more do business on their RESTful APIs.

G. Saldamli · A. Doshatti · D. Kapadia · D. Nyati · M. Bodiwala
Computer Engineering Department, San Jose State University, San Jose, CA, USA
e-mail: gokay.saldamli@sjsu.edu

L. Ertaul (✉)
Department of Computer Science, California State University East Bay, Hayward, CA, USA
e-mail: levent.ertaul@csueastbay.edu

So the next question can be, what is RESTful API and what does it do? If thought at a very basic level, API is going to control what data will the front-end show and what operations need to be done on that data. The data can be static but most usually API communicates with the database to manipulate on the data. To understand it in the real world, let us take an example of Airbnb. The most basic thing a user do on Airbnb is searching for a property and when a user requests the application to show him/her properties in a particular region and in a date range, then that information is sent to the API which manipulates the database to find all the properties which satisfy the user's requirements. This API uses the GET method of Hyper Text Transfer Protocol (HTTP) since it is just getting the data from the database. But one can think, how was the database populated in the first place. For that, API which uses POST method of HTTP is used by the front-end to post a new property into the database. Not just that, using PUT method, APIs can edit the already stored data in the database, that is, for changing the price of already posted property. To remove a property, DELETE method is used by the API to remove that record from the database. This was an example of Airbnb, but the same things can be visualized in other applications too like Grubhub, Facebook, and many more. So if thought at a lower level, it all comes down to interacting with the database using queries and the normal CRUD (Create, Read, Update, and Delete) operations are a must when it comes to developing any web application.

In the world of start-ups, the need for fast development of applications is rising. Companies constantly urge developers to create applications and other business software more quickly without sacrificing quality. In such situation, having a solution which provides RESTful APIs on the go based on the database details can be seen as a blessing. To address this situation, this paper provides a solution by offering enterprise back-end as a service [1]. The solution discussed in this chapter communicates with the user to get their requirements in terms of database and takes care of the tasks like creating the database, hosting the database and creating a CRUD-based RESTful APIs, and providing the code for the same. Along with that, the solution also provides machine learning-based RESTful APIs for the enterprise businesses to get insight into their data. Small organizations or start-ups face problems in creating back-end APIs and hosting the server on cloud platforms. Automated generation of back-end APIs can save a lot of time. It can benefit not only IT experts but also non-IT people. It can help back-end developers, front-end developers, data scientists and nontechnical people as follows:

- “Back-end Developers”: It can help back-end developers to create databases, quickly create APIs by auto generation, modify the APIs if needed and quickly host the server on some cloud platform.
- “Front-end Developers”: It can save a lot of time for front-end developers who need back-end APIs and databases hosted on a cloud platform.
- “Data Scientists”: It can help data scientists and machine learning engineers as they can quickly save data into databases using APIs, add on the APIs for their purpose.

- “Nontechnical People”: It can be a lifesaver for nontechnical people as they can create databases using this, create and host a server with RESTful APIs, connect with a machine learning service. Later, they can even use the code to extend the functionalities.

2 Related Work

A key factor driving business growth is the need to minimize nuances in the development of applications. Backend as a service reduces the subtleties of mobile and web applications development and design. It eliminates the need for application developers to create their own server interaction process back-end system. Backend-as-a-service vendors offer solutions that do not require complex coding for server hosting, which reduces application development time and improves front-end tasks such as application design and user interface (UI) design [1]. In the paper, “Availability Evaluation and Sensitivity Analysis of a Mobile Backend-as-a-service Platform,” Costa, Igor, Jean Araujo, Jamilson Dantas, Eliomar Campos, Francisco Airton Silva, and Paulo Maciel have explained how back-end as a service enables engineers to connect their application back-end to cloud servers. They also explain, how back-end as a service can be used to integrate with other apps [2]. After the understanding of back-end as a service, we dig deep to understand different types of structures in databases which are the integral and most important parts of any system if you are focusing on database. In this chapter, having a look toward a REST-based Universal API for Database-as-Service Systems,” Till Haselmann explained that the objectives of the API should be to provide maximum flexibility and exchange ability, for which the relational databases operate as the greatest source and we should, therefore, use the SQL database. To support this thought, we found that in “A relational database environment for numerical simulation back-end storage,” Jacek Nazdrowicz also supports the thought of using SQL databases for back-end [3, 4]. A few efforts have been made to understand relation between Model-Driven Engineering and Web Engineering which normally called Model-Driven Web Engineering which proposes the utilization of the models and model changes for the specification and semiautomatic age of web applications [5–7]. Most of the projects have essentially utilized information models, route models, and presentations models to automate the process of the REST API creation [8–10]. Most of them are providing features of creating web services but generating the RESTful APIs is very less and even if they are providing APIs the approach require us to model the APIs in specific DSL from which it generates the APIs. Going forward one more step, EMF-REST [11, 12] approach generates RESTful web APIs from EMF models. This implementation has filled the gap between modeling methodology and web technologies. The Eclipse modeling framework also has one limitation that it does not provide the code which has its own limitation of maintenance. There are also several approaches explored about generating REST API from a legacy application which follows a common process of reverse engineering

of L-System to RESTful APIs [6]. It is also having the same limitation they are not providing any code which comes with one more limitation of maintenance and future changes. This work generates the RESTful APIs from the MODELS which is also one of the adaptations of model-driven approach, with an added advantage of the access to the automatically generated REST APIs, which helps the stack holders to deal with the real code and makes the changes based on their requirements. Code generation as a service uses Epsilon [13] to perform model-to-text transformations, the generated output can be utilized for any language, and the service is being implemented with an API, so any client program can do the maximum use of it [14], which is another improvement of this study is doing. Our work is removing the requirement of Epsilon and also providing a complete code in proper folder and file structure which can be utilized or maintained by the client.

3 Overview

3.1 Problem Statement

As explained in the above section, back-end is a middleware that handles the functionality of an enterprise application via API or SDK. Backend as a service allows users to maintain only the front-end with everything behind the scene aspects related to the back-end being managed by the service model.

Our problem statement is to build an application that will interact with the user to create a back-end code that should create the required product with bare minimum requirements starting from the creation of the database to the creation of the APIs to communicate with the database. The user should be able to build applications in a few minutes with just a few clicks. The user should be able to achieve three main goals:

- **Build Application:** The user should build the back-end applications with just a few clicks and be able to host them on different cloud services.
- **Generate Code:** The user should be able to generate the back-end code without having any prior knowledge of coding.
- **Build Databases:** The user should be able to migrate their existing databases or create new databases easily using our application.

3.2 Architecture

The architecture can be divided into three parts. The first part is the user interaction part as shown in Fig. 1. Here, we would have a website wherein user would input data in forms. The user would be able to give their specifications and details using these forms. After collecting the input in the second part, we have the EBaaS server,

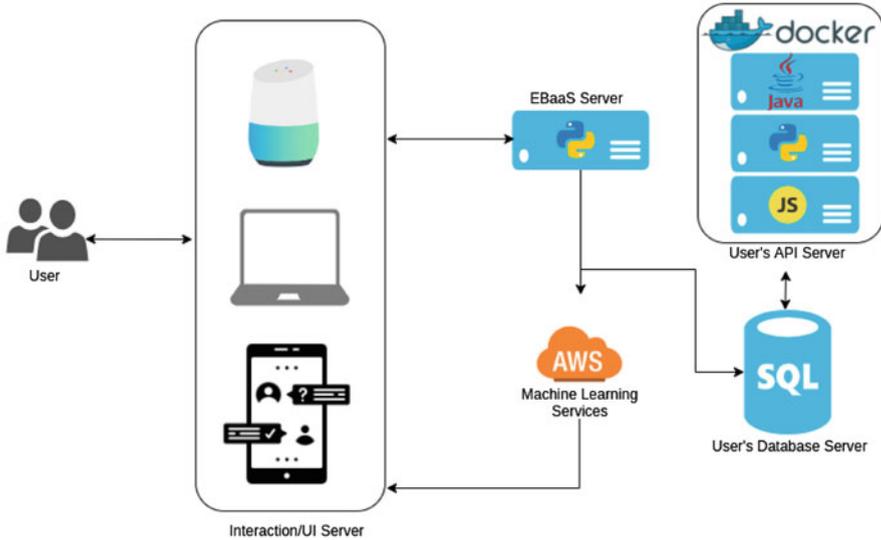


Fig. 1 Architecture diagram

where our code would process the inputs and generate the database with the schema on the user provided database server and generate the code for APIs for the same and publish a Docker image with the code for the code.

We plan to use the machine learning services provided by AWS to suggest the data points which user should consider adding to their schema to make it more good. The output would be a published Docker image which the user can download and spawn a container out of it on its API server. The container will also have the code, so the user can get into the container and customize the code if he/she wishes to do so.

4 Methodology

This section of the paper describes the process that goes behind bringing the idea of Enterprise Backend as a Service into existence. Our study aims for allowing user to control the information from nothing but just the mere knowledge of the user of what information he/she wants to manage. Putting it in a technical context, this study is trying to provide the user with a back-end that gives control to the models which are nothing but a representation of the information the user wanted to manage. Breaking it into various processes, two major processes stand out: (1) generating the models from the information the user provides and (2) generating the actual back-end to control those models.

For simplicity, let us discuss a use case where the user would feel the need to use our solution and how the solution works for that particular use case. Consider a situation where a user is tasked with building a prototype of an employee management system within an organization in a short span of time. Now, the user would be needing models which can then be translated into a database system and obviously an application which can handle basic operations like creating, retrieving, updating, and deleting of resources of a particular model. This will be the first and the most basic requirement that the user would have to kick start the more complex operations on the models. In fact for that matter, any model-driven system would need an application which can handle the basic CRUD (Create, Retrieve, Update, Delete) operations as a starting point. Now, in this particular situation, the user could either opt to design the application that handles CRUD operations which could take up a substantial amount of time for something very basic but important or opt for a solution which could provide this basic application on hand with nothing but the information about the model. This is where this work comes into the picture to make life easier for that user and saving the user's days of designing.

Before digging deep into the major processes discussed above, let us discuss about the actual application that the user would interact with for obtaining results. The application runs on a load balanced EC2 instances on Amazon Cloud Service which is hosting the application's front-end developed in JavaScript and ReactJS framework and back-end which is developed using Python and Flask Framework. The user would be presented with a login screen on a start-up and hence allows the application to manage each and every user's projects separately and securely from one another. This calls for the need of the database which is an SQL database hosted on Amazon Web Service's RDS service. Coming to the security, the actual application never really stores the database records anywhere within the whole system and only helps users to create a database in the first place. After the creation, the user has full access to the database without the actual application having any access to that. So, now it is time to dig deep into the major processes.

4.1 Generating Models

As discussed above, the first thing the user would want in this use case would be the models which can be mapped to and from the database system. The actual application never gets too harsh with the user which can be proved by the way the application asks the information from the user. Let us say that the user in this case has no idea of what different kinds of information the organization wants to manage except for the employee's name and salary. To handle such cases, the application presents user with the user interface which asks information progressively (Fig. 2).

The very basic thing the user would need over here is the server where the database can be hosted. This again brings forward the point that once the database and the back-end are created, none other than the user will have access to that database's records. So, the application will put forward four options to the user

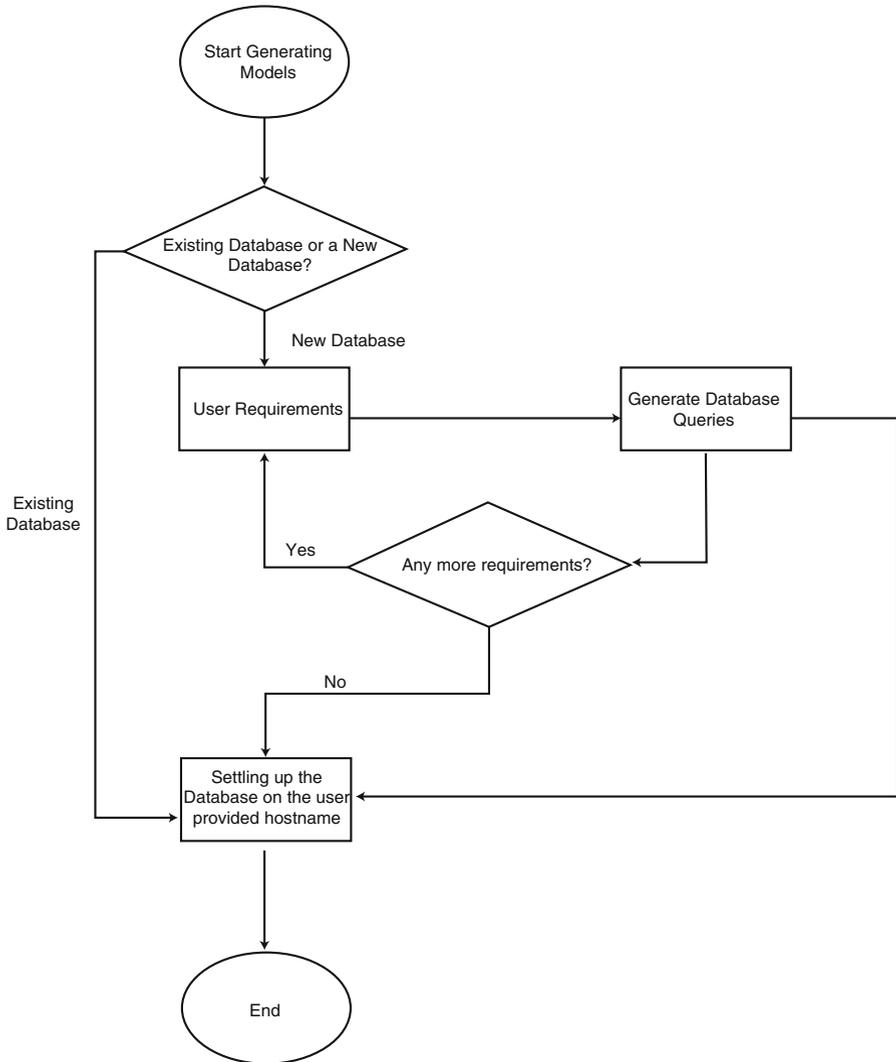


Fig. 2 Generation of models—workflow

allowing him/her to connect to the database, namely: (1) create a new database, (2) connect to an existing database, (3) connect to a database using Excel, and (4) connect to a database using SQL file. In this case, let us assume that there is no existing database that the user has and selects the first option of creating a new database. As shown in Fig. 2, the application would ask user to enter details like host name, username, password, database name, and connection name. Connection name is required to extinguish multiple connections to various databases the user connect to. This would result into a database creation on the server hosted by host

name. Did not that feel like a magic? Behind the scenes, this operation of the user would result in a request hitting the application’s back-end which does nothing but executes the required commands to connect to the host and creating a database. User has decided to create a new database called “test” locally denoted by localhost as the database address in Fig. 3. On submitting the request, the application would create a database named “test” with privileges given to the user “root” with no tables which can be seen in Fig. 4.

Now, since the user in this case has created a new database, there would not be any tables present in that database. Hence, the next thing the user interface presents to the user is a chance to create tables. As discussed earlier, the user only knows about managing employee’s name and salary and hence will suffice with creating only one table and on successful creation, the user will progress to add columns like name and salary.

These operations would be presented to the user as shown in Figs. 5 and 6. Figure 5 allows user to add tables and Fig. 6 allows user to add columns to the tables which

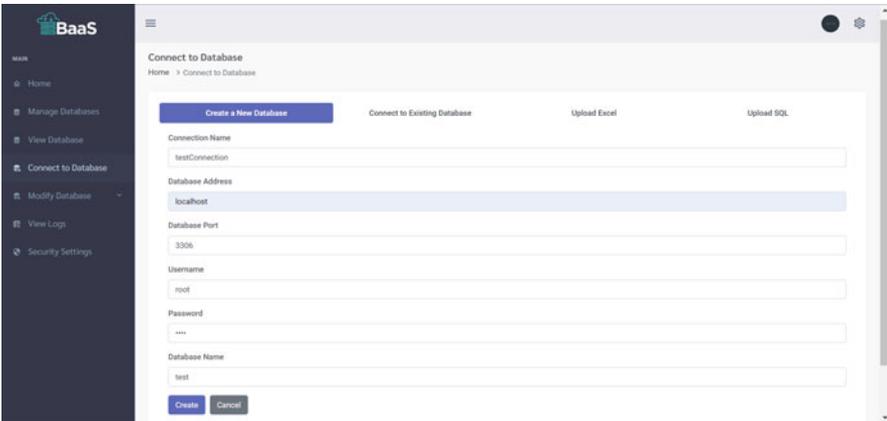


Fig. 3 Create database form

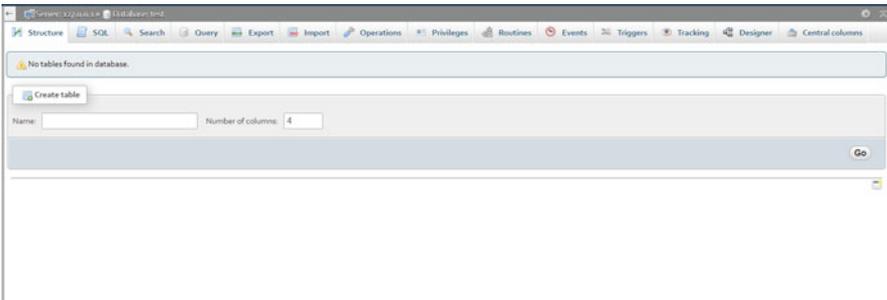


Fig. 4 Corresponding database with no tables

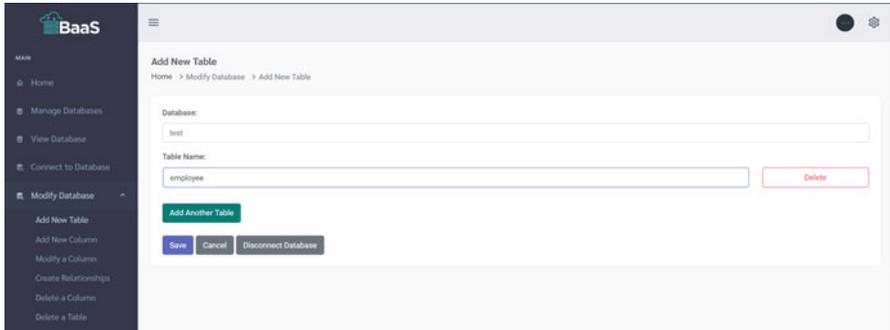


Fig. 5 User interface to add tables in a database

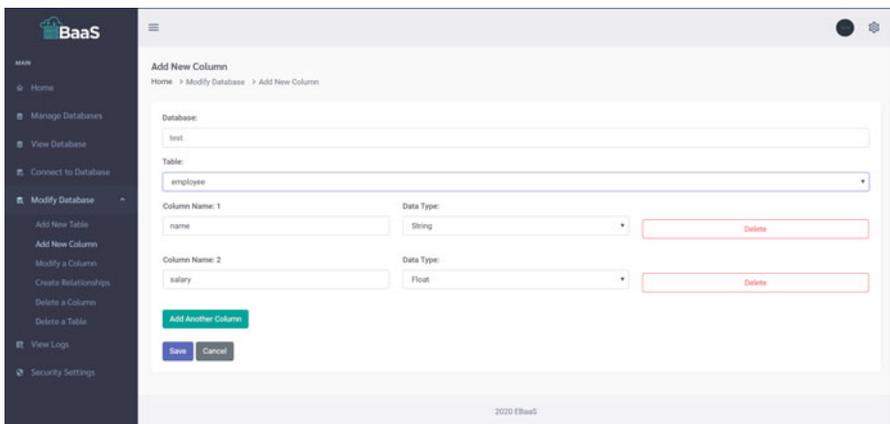


Fig. 6 User interface to add columns in a particular table

the user selects from the drop-down of already available tables in the database. The corresponding change could be validated in Fig. 7.

By doing this, the user has provided the minimal requirements that it needs to launch an operation of creating a back-end. But what if the user is now informed that the organization wants to manage employee’s addresses too? In that case, the user interface which has been designed for entering information progressively allows user to add a new table into the database and along with that establishing a relationship between those two tables. This again, behind the scenes would result in operations that execute SQL queries to reflect the user changes on the actual database.

This way the application generates the database model step by step. There is no groundbreaking technique which the application is using here to generate these models but instead is focusing on operating and manipulating database models through simple SQL queries known to the whole world in a systematic and a more structural way. For example, the user’s request to add a new column to the table

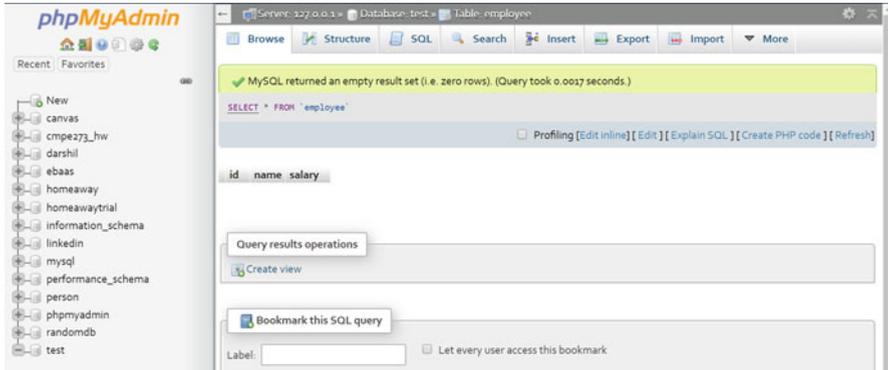


Fig. 7 Corresponding change in database

with specific data type would do nothing but trigger an “ALTER TABLE” query behind the scenes. The progressive way of asking the information from the user will make sure that the table “ALTER TABLE” is trying to alter already exists as the user would have been asked to create a table first before adding columns.

This database model will help in generating object-oriented models which our back-end will be using to manipulate data within the database. Hence, the next step would be to map models out of these already created tables and generating the code to manipulate database through those models. That is where our next process of “Generating Code” comes into the picture.

4.2 Generating Code

Generating code is something that has brought that extra edge to this propose work. To be even more specific, this work aims to generate a back-end code with basic CRUD operations on the models. The first decision that had to be made over here is deciding over the service’s architectural style. With many architectural styles out there like SOAP, REST, RPC, GraphQL, etc., the style which has dominated the market in the past decade has been the RESTful architectural style. The support of REST style is ever increasing and competence of using REST is something which every programmer is demanded for. Not just that, majority of the web services in the past decade has been designed using RESTful architectural style. This was the major reason that drove the idea of generating a RESTful API as a back-end that will further manipulate with the database/models as it allows us to target the maximum crowd out there. Once that decision was made, the next decision was to decide upon the programming language that the code will adhere to which in this case was decided as JavaScript.

Coming to the process of generating code, it is further divided into small processes like deciding a generated code’s file structure, mapping database system to JavaScript models, database connection, type of URIs, etc. Let us discuss these processes one by one.

1. *File Structure:* Once the technology was decided as JavaScript, NodeJS was the clear option to go with as we were creating a server which the client could use to interact with the database. According to [15], the solution decided to organize and present the files around features and not roles (Figs 7 and 8).

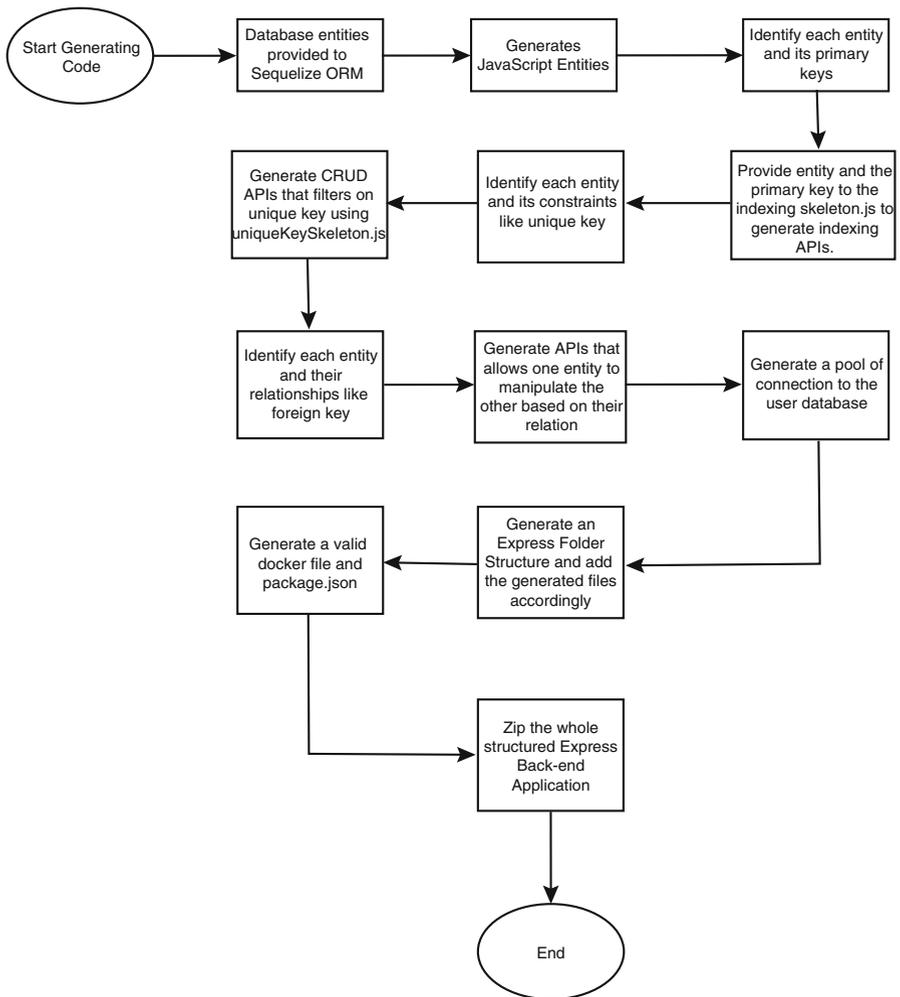


Fig. 8 Generation of code—workflow

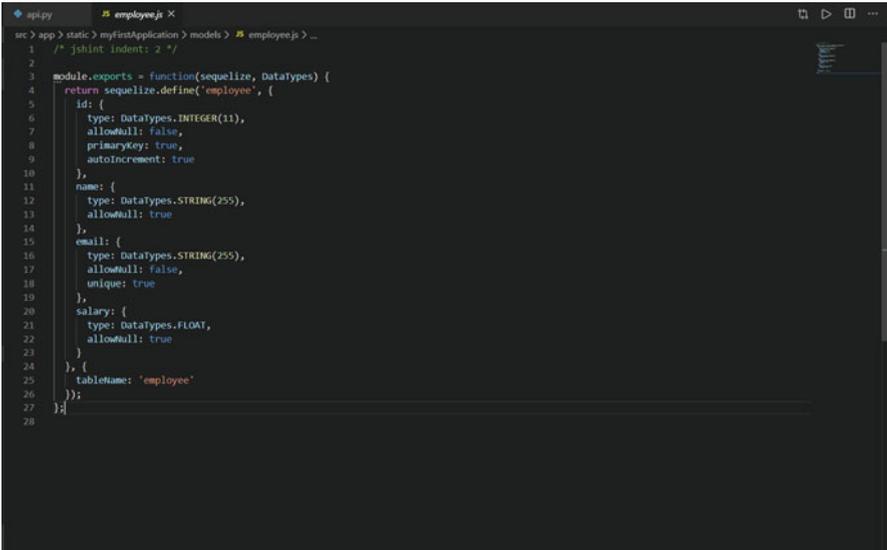
Apart from that, a configuration file that would take care of all the configuration written in JSON and imported wherever necessary. This certifies the uniformity across the back-end code as any change in the configuration could be easily be made by changing only one file in the whole application. Lastly, the logic of each model is separated in the “routes” folder with “index.js” working as the router that routes to the correct file.

This file structure ensures the best practices that are followed and brings modularity in terms of structure.

- 2. *Mapping Models:* Once the user submits the information through the application and wishes to launch an operation of generating the code, the mapping of database to JavaScript models becomes important. Models work as abstraction between the object-oriented programming and the database which is generally called Object Relational Mapping (ORM).

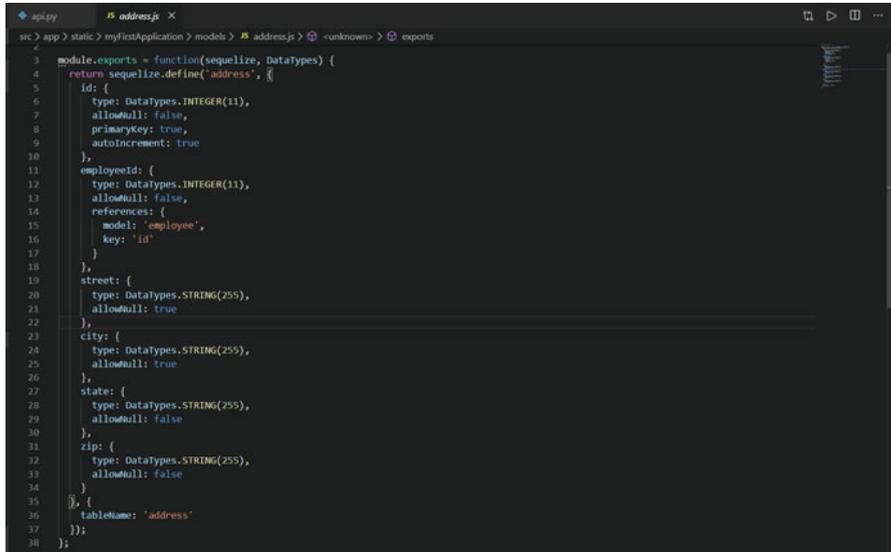
This study uses sequelize as the ORM and exploits the sequelize’s feature of autogenerating the models through specified database using “sequelize-auto” command. Sequelize takes care of converting the database tables into the appropriate JavaScript object models. The table “employee” and “address” in the database has been converted to the JavaScript models as shown in Figs. 9 and 10.

- 3. *Database Connection:* Coming to the connection to the database, our work manages a pool of connection which eliminates problems related to connections like open connections. The configuration of the database like host name, username,



```
1  /* jshint indent: 2 */
2
3  module.exports = function(sequelize, DataTypes) {
4    return sequelize.define('employee', {
5      id: {
6        type: DataTypes.INTEGER(11),
7        allowNull: false,
8        primaryKey: true,
9        autoincrement: true
10     },
11     name: {
12       type: DataTypes.STRING(255),
13       allowNull: true
14     },
15     email: {
16       type: DataTypes.STRING(255),
17       allowNull: false,
18       unique: true
19     },
20     salary: {
21       type: DataTypes.FLOAT,
22       allowNull: true
23     }
24   }, {
25     tablename: 'employee'
26   });
27 }
28
```

Fig. 9 Employee model mapped from database



```
1  module.exports = function(sequelize, DataTypes) {
2
3    return sequelize.define('address', {
4
5      id: {
6        type: DataTypes.INTEGER(11),
7        allowNull: false,
8        primaryKey: true,
9        autoIncrement: true
10     },
11     employeeId: {
12       type: DataTypes.INTEGER(11),
13       allowNull: false,
14       references: {
15         model: 'employee',
16         key: 'id'
17       }
18     },
19     street: {
20       type: DataTypes.STRING(255),
21       allowNull: true
22     },
23     city: {
24       type: DataTypes.STRING(255),
25       allowNull: true
26     },
27     state: {
28       type: DataTypes.STRING(255),
29       allowNull: false
30     },
31     zip: {
32       type: DataTypes.STRING(255),
33       allowNull: false
34     }
35   }, {
36     tableName: 'address'
37   });
38 }
```

Fig. 10 Address model mapped from database

password, and database name is stored in the configuration file for a single point of manipulation.

- 4. **URIs:** Selecting the URIs that the solution would provide was the most critical part. Our work only kept focus on providing CRUD operations in the generated code. But even then, updation, deletion, and retrieval of the resources need a parameter which acts as a filter on which the operation is going to get performed. For example, if there is an employee with ID as “1” and if the user wants to change that employee’s name, then the URI would be /employee/;id; where ;id; will be replaced by 1 and the change would be made on only that particular employee. Hence, it works as a filter. But ID is said to be an identifier and more often than not this identifier is managed directly by the database. Hence, it cannot be assumed that the user would always know the employee’s ID.

Hence, giving only the identifier as the filter option would have made most CRUD URIs useless for the users and also it would have been a chaos if the project provided each attribute of the model for filtering purpose. Analyzing the situation, a middle ground had to be established. Looking at the employee model, it shows that “email” attribute is unique across all employees which could be exploited by the user for filtering through the employees. This very idea is put into effect in this work where the filter options include identifiers like IDs and unique fields like email, name, etc.

Apart from that, URIs to honor the relationship between models are also provided. For example, one employee model record can hold many records of address model. So the user should be allowed to manipulate the child table (address)

through parent table (employee). The example of one such URI is presented in Fig. 11.

Once the types of URIs were decided, the project digs deep into the database schema to find information about the structure like number of tables, columns in tables, relationship between tables, etc. This information is then provided to the templates one by one to create the actual code from the template. An example of template and its corresponding actual code are shown in Figs. 11 and 12, respectively.

Once the whole operation from generating models to generating code is completed, the application creates a zip file of the generated code readily available for the user to download.

5 Deployment and Maintenance

5.1 Deployment

The architecture diagram briefly explains the components in our work. We can divide the architecture in different planes, namely, the management plane, API plane, and the storage plane. We have to do the deployment and maintenance of our interaction server, which is the management plane in our architecture. User would

```

28
29 router.post('/:id/childtable', function (req, res, next) {
30   var childEntity = Conn.import(__dirname + '/../models/${config.YOUR_TABLE_NAME}.js');
31   let id = req.params.id
32   let body = req.body
33   body['foreignkey'] = id
34   Conn.sync({ force: false }).then(() => {
35     childEntity.create(body, { raw: true })
36       .then(result => {
37         // console.log("entity is: ", result);
38         res.writeHead(200, {
39           'content-type': 'application/json'
40         })
41         res.end(JSON.stringify(result))
42       })
43     .catch(err => {
44       res.writeHead(500, {
45         'content-type': 'application/json'
46       })
47       res.end(JSON.stringify(err))
48     })
49     // console.log("sync is completed")
50   }).catch(err => {
51     res.writeHead(500, {
52       'content-type': 'application/json'
53     })
54     res.end(JSON.stringify(err))
55   });
56 });

```

Fig. 11 Skeleton for designing URI to connect child and parent table

```
mc > app > static > myfirstApplication > routes > # employee.js > ...
246   res.writeHead(500, {
247     'Content-type': 'application/json'
248   })
249   res.end(JSON.stringify(err))
250 });
251 });
252
253 router.post('/:id/address', function (req, res, next) {
254   var childEntity = conn.import(__dirname + '/../models/address.js');
255   let id = req.params.id
256   let body = req.body
257   body['employeeId'] = id
258   conn.sync({ force: false }).then(() => {
259     childEntity.create(body, { raw: true })
260     .then(result => {
261       // console.log("entity is:", result);
262       res.writeHead(200, {
263         'Content-type': 'application/json'
264       })
265       res.end(JSON.stringify(result))
266     })
267     .catch(err => {
268       res.writeHead(500, {
269         'Content-type': 'application/json'
270       })
271       res.end(JSON.stringify(err))
272     })
273     // console.log("sync is completed")
274   }).catch(err => {
275     res.writeHead(500, {
276       'Content-type': 'application/json'
277     })
278     res.end(JSON.stringify(err))
279   });
280 });
281
282 module.exports = router;
```

Fig. 12 Generated code from the skeleton

interact with this as it would be hosting the front-end and rest of the deployments are to be done at the users end.

1. *Management Plane:* For the management plane, we have the management server hosted in cloud on a Docker hub having resilient kubernetes cluster serving the APIs. The management plane is further divided into two servers: the front-end server and the API server, that is, the Docker hub. The front-end server hosts our React application and the API server provides us the APIs.
2. *Storage Plane:* The deployment of the interaction plane would be done after the deployment of storage plane. The storage plane will basically have the database servers which user wants for their application. The user needs to have the sql client server installed on the server and can use any deployment strategy and storage strategy for their database. The user just needs to provide the important credentials of the database to create tables in the database and do all the important changes required in their database and get APIs for content of database.
3. *Interaction Plane:* The interaction plane is the plane in which the user will host the APIs generated by us. Once the user has connected his database with the management plane and launched the application, the completed code for the APIs would be ready and also the Dockerfile required for the same would be ready. The user just has to download the zip and launch the node application or create an image of the Dockerfile and spawn a container of that Dockerfile. To deploy the interaction plane, user can decide his own strategy and device his own architecture. User can have single server hosting the APIs in the node app or user can also have kubernetes cluster inside a Docker hub or a basic Docker

container running the APIs; this would be the interaction plane with single server or multiple server deployed by the user (Fig. 13).

5.2 Maintenance

The responsibility of deployment of management plane is with us and so will be the responsibility of maintaining the management plane is ours. For the maintenance of management, we need to keep the servers always reachable up and running so that we always have a user input. To maintain the management plane with every bug fix or new change, we will have to update the code on the front-end server if we have changes in front-end. For the other changes in APIs, we will have to build a Docker image with the latest code and change the pods in the kubernetes cluster to use the updated Docker image.

The other planes are deployed by the user, so the maintenance is also user's responsibility. For the storage plane, once the user has deployed it with a required architecture, user can make all the changes in the schema of database using the management UI as we can connect to the database and make any changes required in the database. For the maintenance of the interaction plane, if the user wants to change the database or redeploy the APIs or make any changes in API code, he just needs to make the required change himself. So the steps to keep the interaction plane updated are the user need to make the changes in database using the UI and download the latest code once the code is downloaded and updated by making any changes in code the user must again the deploy code in the server and start the API server or generate the Docker image of the latest updated code and use the Docker image to spawn containers or in the kubernetes cluster according to the user requirements.

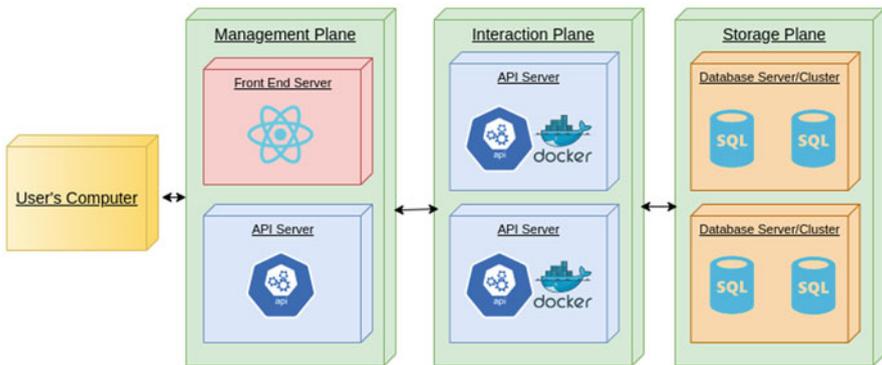


Fig. 13 Deployment diagram

6 Evaluation

6.1 Evaluation Methodology

The main goals of our application are building the application, building databases, and generating the code. We have evaluated our application on these parameters by using different methods. We followed the following methods:

1. We gave our application to different professionals like back-end developers, data scientists, and different nontechnical people and collect the evidence for evaluation. The users evaluated our application on the performance quality. They filled a grid of specifications for the performance. Table 1 represents the mean of all the users' values. We gave the following metrics:
 - Usability: The user used our application for their particular use case. We gave the steps to some users and take their evaluation. The user connected/built the database, generate code, and host the application.
 - Speediness: The users evaluated this metric by how fast they were able to achieve their use case. Evaluating this part was a little tricky. If the user started building the database and then built the application, then the speed will slow down. But if the users connected to an existing database and then built the application, then it was created in just a few seconds.
 - Efficiency: The users evaluated this metric by how efficient our application is. The user checked how efficient our generated code is. How many lines are present. How good the code structure is. Are all the CRUD operations that the user want in our code. How bug free our code is and how RESTful it is.
2. We compared our application with the existing code generators. We compared them with xmysql [16], sand-man [17], and util-raml-code-generator [18]. We had a predefined metrics for this evaluation also. We evaluated all the applications on the following features:
 - Code quality: Once the code is generated, we evaluated how bug free the code is, how good the file structure is, how many lines of code are present, and how accessible, editable, and modifiable the code is.

Table 1 Evaluation of Usability, Speediness, Efficiency, Code quality, Database connectivity, and Cloud services

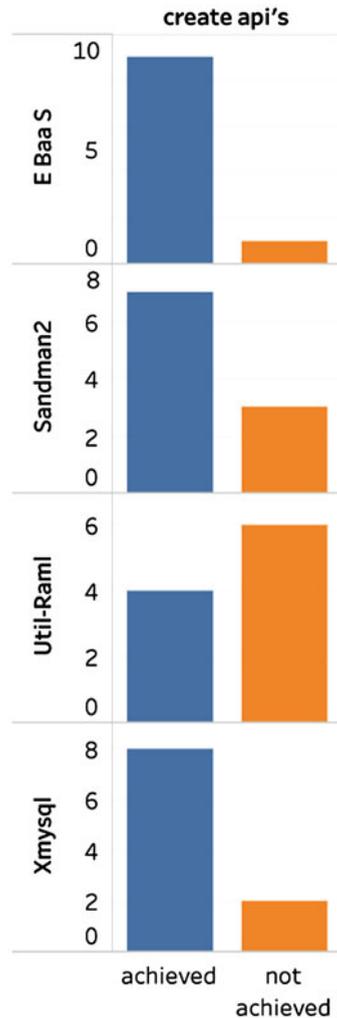
	util-raml	Xmysql	sandman2	EBaaS
Usability	2/5	3/5	3/5	5/5
Speediness	2/5	5/5	5/5	4/5
Efficiency	3/5	2/5	2/5	4/5
Code quality	4/5	N/A	N/A	4/5
Database connectivity	N/A	N/A	4/5	4/5
Cloud services	N/A	4/5	4/5	5/5

- Database connectivity: We evaluated how easy it is to integrate the users existing database or create new databases, edit tables, and create complex relationships.
- Cloud Services: We evaluated how easy it is to host the code provided these applications.

6.2 Performance and Benchmarks

- Usability: Ten users tried to create rest APIs using all the four applications. Nine users were able to successfully create the APIs using EBaaS. Figure 14 shows

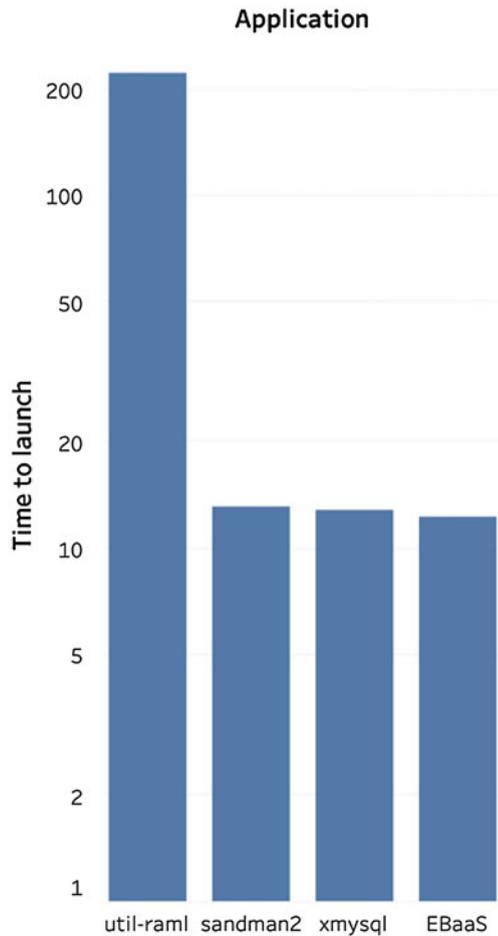
Fig. 14 Usability to create APIs



the comparison of all the four applications. Util-raml was a little complicated as the user had to create the raml file. Four of 10 users were able to generate APIs using Util-raml. Sandman2 and xmysql performed quite similarly. Users with prior technical background were easily able to launch the APIs but users with nontechnical background faced challenges. Seven of 10 users were able to launch APIs using Sandman and eight out of 10 users were able to launch using xmysql.

- **Speediness:** We tracked the average times to launch the APIs for different use cases. The use case to connect to an existing database was common for all the applications. Util-raml performed quite slow as the users had to create a raml file for their database and give that file to the code generator. It took around 4.3 min to create APIs using util-raml. Xmysql, sandman2, and EBaaS performed

Fig. 15 Time to launch an application



had nearly the same speed. It took around 12 seconds to launch the applications. Figure 15 shows the comparison between different applications.

- Code quality: xmysql and Sandman2 do not give code. So we did not add them in this comparison. Util-raml and EBaaS both performed pretty good. For the same use case, util-raml had 102 lines of code for entities and 166 lines of code for services, while EBaaS had 26 lines of code for entities and 143 lines of code for services. Both the codes generated were quite modifiable. Util-raml gave the code in PHP and JavaScript while EBaaS generated the code in JavaScript. Figure 16 shows the comparison between the different applications on the basis of lines of code.
- Database connectivity: EBaaS performed the best here. EBaaS supported database connectivity from existing databases, and also gave access to create new databases from scratch or by uploading a sql file or Excel file. It was difficult to connect to database in util-raml because the user had to create the raml

Fig. 16 Lines of code

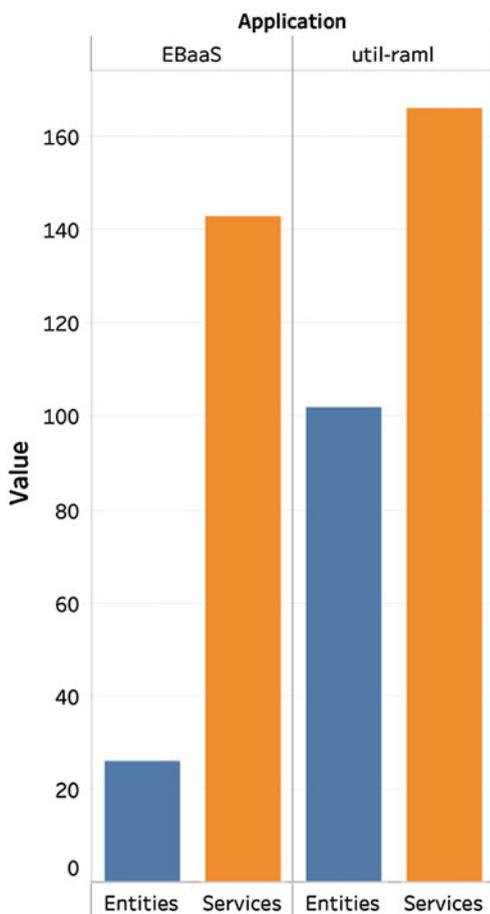
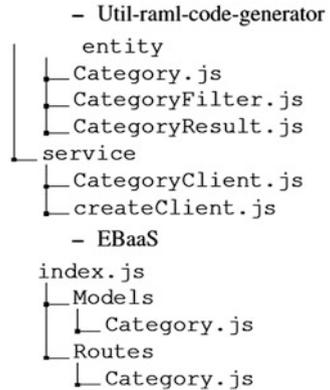


Fig. 17 File structure comparison



files. xmysql and Sandman2 take the database string in the parameters. They do not give features to create a new database or modify an existing database. Sandman2 supports many different types of databases. EBaaS only supports MySQL databases (Fig. 17).

- Cloud services: The world is quickly moving toward cloud computing which allows software to be available based on demand. To make it even more feasible, containerization has taken its own place allowing to provide better security, portability, fast deployment among other benefits. Keeping this in mind, EBaaS provides a Docker file on hand which provides users to easily make their back-end application containerized. Even though Xmysql and Sandman2 provide Docker support, Util-raml-code-generator fails to do so.

7 Conclusion

EBaaS is basically a one stop application for developing and maintaining the complete back-end of any application. Interacting with the UI, user can develop the database, make updates in the database, and have the API code ready to integrate the APIs with their own front-end. User with their own servers can have the complete, efficient, and speedy back-end with all basic APIs ready with just few clicks and inputs. User can later make any changes in the API code provided by our code generator to also have user-defined APIs. So EBaaS basically as the name suggested is an application to provide enterprise backed as a service.

8 Future Work

Our work currently focuses on the connection/creation of the relational databases based on which it is generating the RESTful APIs. In future, the scope of this work can be expanded to the non-relational databases. This work has so much scope of incorporating ML services which can be used later on for recommendation based on the use case. For example, if the use case involves sales, then a ready to use ML API service to gain insights into the sales and much more. Currently, this study is reserving REST web architecture for generating the back-end code, it could be made even more useful by giving users the option of selecting different architectures like GraphQL, SOAP, RPC, etc. When it comes to automation, the work, research, and innovation are never going to stop and hence the future capabilities of this work can also be infinite.

References

1. Technavio updates on the global back-end as a service market. Professional services close-up. Online available: <https://advance.lexis.com/api/document?collection=news&id=urn:contentItem:5TVM-M141-F06S-P1FM-00000-00&context=1516831>
2. I. Costa, J. Araujo, J. Dantas, E. Campos, F.A. Silva, P. Maciel, Availability evaluation and sensitivity analysis of a mobile back-end-as-a-service platform. *Qual. Reliab. Eng. Int.* **32**(7), 2191–2205 (2016)
3. T. Haselmann, G. Thies, G. Vossen, Looking into a REST-based universal API for database-as-a-service systems. *2010 IEEE 12th Conference on Commerce and Enterprise Computing*, Shanghai, 2010. pp. 17–24, <https://doi.org/10.1109/CEC.2010.11>
4. J. Nazdrowicz, A relational database environment for numerical simulation back-end storage. *2015 22nd International Conference Mixed Design of Integrated Circuits & Systems (MIXDES)*, Torun, 2015. pp. 601–606, <https://doi.org/10.1109/MIXDES.2015.7208595>
5. S. Ceri, P. Fraternali, A. Bongio, Web modeling language (WebML): A modeling language for designing web sites. *J. Comp. Netw.* **33**, 137–157 (2000)
6. R.T. Fielding, *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis (2000)
7. N. Koch, S. Kozuruba, Requirements models as first class entities in model-driven web engineering. In *ICWE Workshops* (2012), pp 158–169
8. X. Qafmolla, V.C. Nguyen, Automation of web services development using model driven techniques. In *ICCAE Conference*, vol. 3 (2010), pp. 190–194
9. A. Schauerhuber, M. Wimmer, E. Kapsammer, Bridging existing web modeling languages to model-driven engineering: a metamodel for WebML. In *ICWE Conference*. (2006)
10. WebRatio, <https://www.webratio.com/site/content/en/web-application-development>. Accessed April. 2020

11. H. Ed-douibi, J.L.C. Izquierdo, A. Gómez, M. Tisi, J. Cabot, EMF-REST: generation of RESTful APIs from models. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing (SAC'16)*. Association for Computing Machinery, New York, NY, USA (2016), pp. 1446–1453
12. Liu, Y., Wang, Q., Zhuang, M., Zhu, Y., Reengineering legacy systems with RESTful web service. In: *2008 32nd Annual IEEE International Computer Software and Applications Conference (2100219007)* (2008), pp. 785–790
13. D. Kolovos et al., Epsilon (Jul 2015), <http://www.eclipse.org/epsilon>
14. R. Crocombe, D.S. Kolovos, Code generation as a service. In *CloudMDE@ MoDELS* (2015, September), pp. 25–30
15. <https://blog.risingstack.com/node-hero-node-js-project-structure-tutorial/>
16. <https://github.com/o11ab/xmysql>
17. <https://github.com/jeffknupp/sandman2>
18. <https://github.com/paysera/util-raml-code-generator>



Aspen Olmsted

1 Introduction

Forrester Research defines business intelligence as “a set of methodologies, processes, architectures, and technologies that transform raw data into meaningful and useful information used to enable more effective strategic, tactical, and operational insights and decision-making” [1]. For today’s businesses, this mainly takes shape through data visualization (tabular and charts), business documents, data mining, customer interaction automation, and email marketing.

Data visualizations have been developed by enterprises for decades to allow users to analyze their data in tabular or chart format. The visualizations change based on runtime prompts that filter the data displayed in the visualization. Data from separate Online Transaction Processing (OTP) systems are often aggregated into data warehouses to allow visualizations that span data from multiple source systems. Unfortunately, little tooling was provided to ensure the data visualizations guaranteed the required availability, integrity, and privacy required by the organization. In our previous work, we developed an XML language to test the availability of business intelligence documents executed from the cloud [2]. This paper describes our enhancements to that work in a platform we call Secure Business Intelligence Report (secBIrpts). Our platform secBIrpts allows an organization to script the privacy requirements into the configuration, and the platform will enforce the rules through a Mandatory Access Control (MAC).

Traditional reporting engines either utilize a single set of credentials for all users with access to a report or pass the operator’s credentials through the business intelligence platform to the back-end database. Unfortunately, the confidentiality of the

A. Olmsted (✉)

Department of Computer Science, Fisher College, Boston, MA, USA

e-mail: aolmsted@fisher.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_79

1101

source data is often ignored, leading to knowledge transfer that violates organization policies. Utilizing the secBIRpts language, an organization can configure the privacy requirements into an Extensible Markup Language (XML) document. The XML document will store the hierarchy of labels that are applied to data retrieved via database tables, views, and stored procedures, along data retrieved from external sources via web services.

Our original client-side testing language would ensure the integrity and availability of data mining and automation emails. Data mining allows an enterprise to discover new knowledge from their OTP data using data science algorithms. Unfortunately, the integrity of the source data is often ignored, leading to new knowledge derived from incorrect information. Utilizing secBIML, an organization can script the correctness requirements into comparison tables and receive proactive notification of integrity issues in the source data. The data sources in secBIRpts can consume the content of the output of the data mining processes and apply MAC hierarchical access control to ensure the privacy of the information that passes through the business intelligence platform.

Many cloud application providers sell customer relationship management (CRM) and email marketing solutions and advertise their ability to automate interactions with customers based on changes in the data. Too little attention is given to how the data are aggregated into the data source for the automation. The availability and integrity of data behind the automation are also often ignored. Our original programming language secBIML can alert an organization of issues so they can proactively solve the problems with the correctness of the data used in the process. We integrated the delivery of the automated emails into secBIRpts to ensure proper privacy along with the integrity and availability guarantees.

Cloud documents such as presentations, word processing files, and spreadsheets often house data from other business intelligence reports. In our original work, we provided tooling in the secBIML language to ensure the correctness of these data. We have extended that work to allow for the automated export of data from secBIRpts to the proper format of the business cloud documents.

Through the integration of the hosting secBIRpts and the client-side testing language secBIML, an organization can ensure the execution of business intelligence while guaranteeing correctness. Neither of the two tools developed in this work, secBIML and secBIRpts, are commercial-grade tools, but they both have been utilized in production environments with strict supervision by the research team.

The organization of the paper is as follows. Section 2 describes the related work and the limitations of current methods. In Sect. 3, we describe the elements in the secBI programming languages. Section IV provides the motivating example behind our work. In Sect. 4, we look at the enhancement to the secBI language that allows the guarantee of privacy in the data consumed in the business intelligence engine. Section 6 describes how we developed our runtime engine. Section 7 drills into the data we gather in our experimentation with data visualizations. Section 8 investigates the tests we used in our experimentation with business document integrity. Section 9 describes the test implementation used in our experimentation

with business email integrity. In Sect. 10, we discuss the increases in security gained by utilizing secBIRpts. We conclude in Sect. 11 and discuss future work.

2 Related Work

The large corporate cloud providers such as Microsoft, Google, Amazon, and IBM hold many patents in the domain of recognizing application availability. The patents are designed for business to consumer websites where there is less control than we have in our enterprise BI environment. The lower level of power stems from the client machines in business to consumer architectures are unknown to the provider. There are many examples of such patents. In such patents, Microsoft inserts a stub between the calling client and the web application. The stub gathers performance data as the user is using the web application. Unfortunately, with such a solution, a flaw in the stub can reduce the availability of the service. In our work, we utilize the network during off-hours for the enterprise to gather application data. The information gathered informs the information technology staff of priorities to proactively solve problems before they are filed as end-user trouble tickets.

Codd [1] describes integrity constraints in his original work on relational databases. Codd's original work assumed the data sources are two-dimensional tables that are normalized to eliminate redundancy. Codd's ideas made it into most online transaction processing (OLTP) databases but never made it to the BI or document level. The data layer behind most BI architectures often increases availability by allowing dirty data through the use of database hints. In our work, we are looking for integrity errors by defining constraints in the document testing language itself and not in the data layer behind the documents.

Many security software vendors offer a web application security scanner. These scanners try to break a web application to find common vulnerabilities such as cross-site scripting and SQL injection. Khoury et al. [3] evaluated the state-of-art black-box scanners that support detecting stored SQL injection vulnerabilities. Our work utilizes white box testing to find weaknesses in access control on both the document or data element level.

In the late twentieth-century commercial relational database management systems (RDBMS) with their role-based access control (RBAC) became very popular. Osborn et al. [2] and Nyanchama and Osborn [4] mapped these RBAC systems to MAC systems using the roles and database constraints. In theory, RBAC systems could have hierarchical roles that function similar to the levels in a MAC system. In practice, the roles have not been implemented. Unfortunately, the RDBMS RBACs only control access to the table, view, and column level. Rows level security is also not enforced in the RDBMS RBAC systems. Our work moves the MAC to the BI level and adds a level of abstraction so we can support table, view, stored procedure, column, and row-level security along with web-service data sources.

Sack et al. [5] were awarded a patent on a database appliance that implements MAC based on the session label and the database object level. The patent does

not define the types of database objects and does not include row-level security. The patent does not handle hierarchical levels. Each context label and security level is a one-to-one mapping. Our work not only implements similar database vendor independent features but also applies row-level MAC security along with the hierarchical security levels found in a model such as Bell-LaPadula (BLP).

We utilize the MySQL [6] database system in our experiments behind our BI platform. Similar to most RDBMS, the MySQL system, there is both discretionary access control (DAC) and RBAC capabilities. MySQL has also added a feature that is often attributed to MAC-based systems. The feature allows a security context to be set from the set of granted roles to the user. The system also supports mandatory roles that must always be applied to the security context [7]. In their work, the context can be manipulated by the end user. In our work, the context is specific to the report configuration.

3 Language Elements

The programming languages secBIML and secBIRpts are defined in Extensible Markup Language (XML) with elements expressing the statements and expressions. Attributes or child elements represent the parameters of the statements and expressions. Elements are identified in a SecBIML program as a start tag, which gives the element name and attributes, followed by the content, followed by the end tag. Start tags are delimited by “<,” and “>”; end tags are bound by “<” and “>.” Table 1 shows a breakdown of the tags available in the secBIML and secBIRpts language

Not all tags are available in each language. The overlap was designed to make it easy to migrate a report developed in a different business intelligence language to secBIRpts. The fourth column of Table 1 shows which language supports the tag.

3.1 Statement Tags

The secBI languages’ syntax is made up of declarative statements that define one of eight statement entities: credential, report, execution, parameter, alert, RESTaction, DBaction, and LOGaction. Figure 1 shows an example set of declarations to define a single implementation of a report test in secBIML with two runtime parameters. The parameters are set for a date range of the entire month of July 2019. The following is the set of language elements currently supported by secBIML:

- **Credentials** – The credential tag declares the connection to a report in secBIML. This connection is typically a user and password required by the web page to access the report. In secBIRpts, the credentials tag allows for links to a back-end data connection.

Table 1 secBIML and secBIRpts tags

Tag	Type	Parent	Language
Credential	Statement		Both
Report	Statement	Credential	Both
Execution	Statement	Report	secBIML
Parameter	Statement	Report	Both
Outputtable	Statement	Report	secBIRpts
Outputcolumn	Statement	Outputtable	secBIRpts
Alert	Statement	Comparison or Outputtable	Both
Dataconnection	Statement	Outputtable	secBIRpts
Datajoin	Statement	Outputtable	secBIRpts
Datasource	Statement	Dataconnection	secBIRpts
Datacolumn	Statement	Datasource	secBIRpts
RestAction	Statement	Alert	Both
DBAction	Statement	Alert	Both
LogAction	Statement	Alert	Both
Reference	Expression	OutputColumn or Comparison	Both
Comparison	Expression	Datasource	Both
Literal	Expression	Comparison or OutputColumn	Both
ActionValue	Expression	RESTAction, DBAction, or LogAction	Both

```

<report name="eventbyhour"
server=https://logireports.fi.edu?rdName=Reports.Admissio
ns.Event_ByHour credential="bilogin"/>

  <execution name="eventbyhourjuly"
report="eventbyhour"/>

    <parameter execution="eventbyhourjuly"
name="BeginDate" value="07/01/2019"/>

      <parameter execution="eventbyhourjuly"
name="EndDate" value="07/31/2019"/>
    
```

Fig. 1 Example report, execution, and parameter declaration elements

- Reports – The report tag states the details on the server and the name of a specific report that is tested in secBIML. In secBIRpts, the report tag provides additional metadata about the report, including the title.
- Executions – The execution tag declares a specific test case for a report. This tag is currently only supported in secBIML.
- Outputtable – The outputtable tag declares a specific HTML table that is generated in the results of the report or document created in secBIRpts.
- Dataconnection– The dataconnection tag declares the connection to an RDMS database, Business Document, XML REST, or SOAP-based web service. The

dataconnection tag refers back to a credentials tag that expressed the authentication information required to access the data.

- Datasource – The datasource tag defines the query or calls made on a data connection to retrieve data.
- Datajoin – The datajoin tag defines how multiple datasources are combined. For the purposes of this experiment, we only support equal joins on one or more data columns expressed in the secBIRpts language. This tag can be enhanced in the future to support outer and cross joins.
- Outputcolumn – The outputcolumn tag defines cells in the HTML table output produced by secBIRpts.
- Parameters – The parameter tag declares the runtime values used in the test of a specific execution in secBIML. A parameter can also be used to prompt for values in secBIRpts.
- Alerts – The alert tag defines the data that are tested in secBIML. The alert tag is also used to specify the actions to take on success or failure. In secBIML, the alerts are typically used to test for failure. In secBIRpts, the alerts are usually used after a successful execution. Actions can add tuples to a data store, send emails, or call web services. Parent tags for Alerts can either be comparison entities or execution entities.
- RESTactions – The RESTaction tag defines actions that call to web services. The web services call has the key-value pairs in the delivery.
- DBActions – The DB actions tag defines tuples written to a database table. The key in the key-value pair returned from the ActionValue entity matches with a table column, and the value is inserted in the tuple.
- Logactions – The Logaction tag is used to define values written to a log file.

3.2 *Expression Tags*

Expressions in the secBI languages are entities where the syntax returns one of five different data types: list, boolean, numbers, text, or key-value pairs. Expressions are used to find a specific value in the report output, aggregate a set of values in the report output, express literal values, or define what data are sent to actions. Operators can combine expressions to enable complex relational comparisons. Four expression elements return values in the secBIML language. The four items are reference, literal, comparison, and ActionValue. We document the four elements below:

- References – The reference tag allows for the retrieval of a value from a report either in secBIML or secBIRpts. The values are specified in the output by the hypertext markup language (HTML) ID or a position in an HTML table. The type attribute allows values to be accumulated, counted, or averaged. The selector attribute is used to aggregate the values in a row or column within an HTML table. Selectors are patterns that match against elements in a tree and are the

primary method used to select nodes in an XML document. The secBI languages support CSS Level 3 selectors [8].

- Literals – The literal tag allows the expression of a constant value. Literal tags are used when comparing a value in a report to a static value defined at the time when the test is created in secBIML. Literal tags are used when displaying constant values in an output in secBIrpts.
- Comparisons – The comparison tag allows values to be compared. A comparison tag returns a Boolean value based on the results of the comparison. The comparison tag requires an operator attribute to specify the comparison operation type. There are six supported comparison operator abbreviations: equal (EQ), not equal (NE), greater than (GT), less than (LT), great than or equal to (GE), and less than or equal to (LE). The value in the parenthesis is the abbreviated version of the comparison operator. Figure 2 shows an example of the declaration of a reference to a cell that exists in the last row of a table in the report output. A comparison of a literal value of 23,201 is made to the value on the report, and if the data are different, a REST web service call is made to save the data. By default, actions include the data used in the comparison, the name, the compared values, and a timestamp marking the comparison evaluation time.
- ActionValues – The ActionValue tag allows the delivery and storage of key-value pairs in response to the alert. The type attribute defaults to a comparison name but can be a comparison, reference, literal, or execution result. There are two available values from the execution results. The two values are the HTTP status and the duration of the execution.

3.3 Attributes and Child Elements

In both the statement and expression tags, white space and attributes are allowed between the element name and the closing delimiter. An attribute specification

Fig. 2 Example alert and supporting elements

```

    <reference name="attendancetotal"
    execution="eventbyhourjuly" type="sum"
    selector="#attendance"/>

    <comparison name="totalattendance"
    reference="attendancetotal" literal="23201"/>

    <alert comparison="totalattendance"
    action="writeerror"/>

    <action name="writeerror" restaction="
    http://https.logireports.fi.edu/saveerror"/
    actionvalue="#totalattendance">
  
```

consists of an attribute name, an equal sign, and a value. A child element is a tag fully enclosed between the open tag of another statement or expression and the matching closing tag. White space is allowed around the equal sign. Attributes and child elements in the secBI language syntax specify the parameters in the statements or the expressions. It is possible to express any parameter either through an attribute or through a child element. The expression of a child element allows for more complicated settings, including collections of values. Figure 2 shows how the RestAction and ActionValue entities can be rolled up into attributes. Attribute parameters are similar to read but do not allow for more than one value of the same attribute type.

4 secBISQL and Motivating Example

To facilitate the usage of the programming language by non-programmers, we developed a version of the language that has the tags stored in a SQL database. The SQL version is called secBISQL. The semantics of the two language implementations, secBIML, and secBIRpts, are implemented in secBISQL. The difference is in how the programming language is stored in the source format. Figure 3 shows an entity relationship diagram (ER) for secBISQL.

secBISQL was developed for The Franklin Institute (TFI) in Philadelphia, PA [9], to allow them to identify availability and integrity errors in their business intelligence operations. In their business intelligence operations, TFI had 120 custom reports that ran in the cloud using a business intelligence tool named Logi Analytics [10]. The custom reports were developed over many years by several different developers. Unfortunately, the end users were experiencing errors and timeouts throughout the day.

In our first iteration, we used secBISQL to measure the security of data visualizations. We followed this iteration up by experimenting with other generated business documents and communications. The documents we tested can be categorized into three primary categories; word processing, presentation, and spreadsheet documents. Each document we looked at had aggregation of values or references to data from business intelligence reports. We also looked at automated emails sent to patrons after activities with the patrons, along with mass emails that were delivered to patrons to market future events.

For the word processing, presentation, and spreadsheet documents, we utilized Microsoft™ Office 365 [3]. Office 365 is a cloud-based software as a service (SAAS) solution for word processing. The URL of the office 365 document is added in the entity object as a “report” entity. The representation of the URL as an entity allows a programmatic reference to the word processing document. Comparisons can be defined to compare individual values in the document to other values or aggregated values in the same document or a data visualization. For example, an invoice document laid out in Microsoft™ Word can be verified to ensure that the columns for quantity and amount are equal to the total column. A spreadsheet

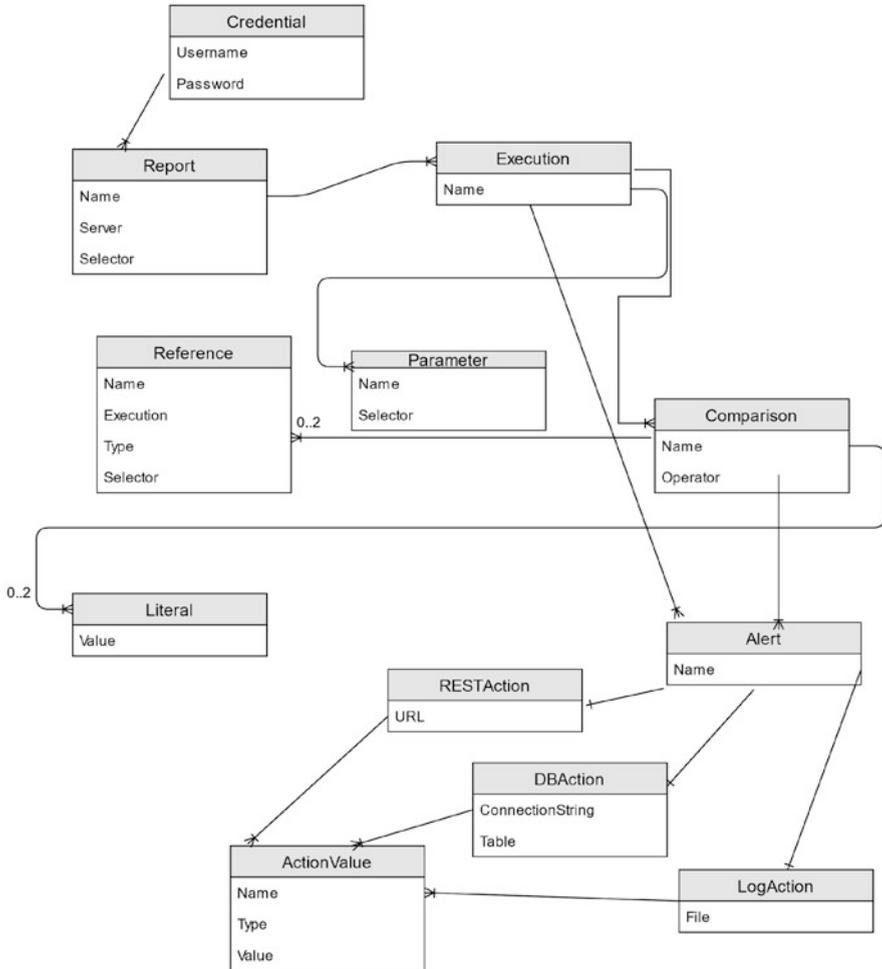


Fig. 3 secBISQL ER diagram

document has the functionality to aggregate values, but a word processing document is often used for the end printed business document because of layout concerns. Integrity checks can be established in secBIML to ensure the word processing data are correct. Values in a business document could also be compared to a source business visualization. Often data are pulled from a data visualization and placed in a flyer or presentation, but that data may change in the source system. secBIML can ensure that data remain correct. This same technique can be used with documents stored in competitive cloud SAAS word processing solution providers such as Google™ GSuite [8] (Fig. 4).

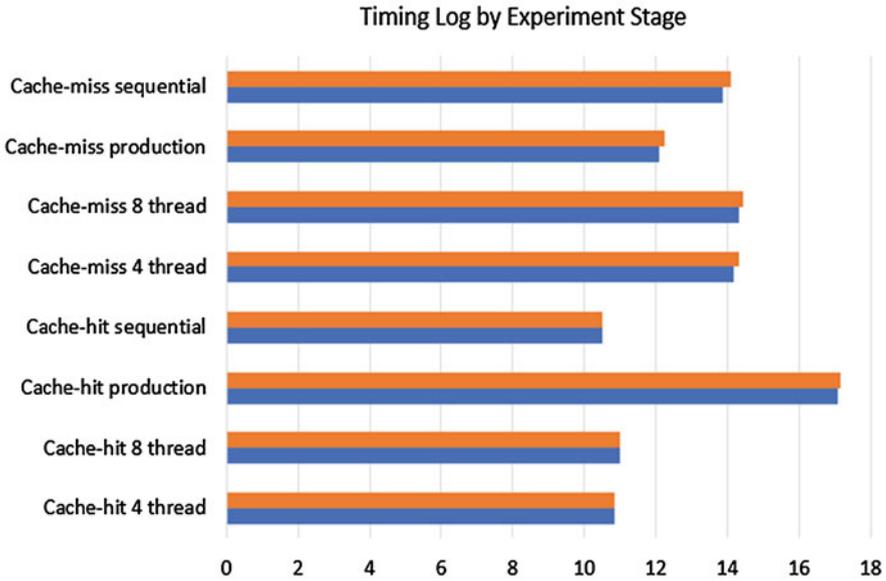


Fig. 4 Average timing

After tackling the business documents, we looked at emails generated from back-end business transactional data. We were able to retrieve emails from an email service provider (ESP) through the representational state transfer (REST) application programmer interface (API)s. REST is a software architectural style that defines a set of constraints for web services creation. Web services that conform to the REST architectural style, called RESTful web services, provide interoperability between computer systems on the Internet. The “report” entity was used to specify a REST front-end URL, and the parameters were used to call out to the web service for the specific REST data. The data were then compared to a report that listed the source data consumed in the generation of the email marketing or business automation.

The final phase of the project was to experiment with replacing the business intelligence platform with the secBIrpts enhancements to the language. The goal of moving the reporting engine completely over to secBI is to add a layer of data privacy guarantees.

5 Implementing Data Privacy

To implement a hierarchical MAC system in the secBI languages, we need to tag data from the source systems to express the privacy levels. We choose to apply an expanded XML Schema Definition (XSD). Most of the data are coming from either

relational database management systems (RDMS) or web services. WSDL web services are compatible with the XSD format and allow a consistent implementation across the different data source types.

We designed a simplified motivating example database for experimentation. Figure 5 shows the ER diagram that includes a patron’s ticket and donation transactions to a museum similar to the Franklin Institute. The tickets and donations entities are the primary sources for the hierarchical privacy model. Each data source was converted into the XSD file, and a virtual element can be wrapped as a complex type around a field used to filter the data by the privacy levels. Figure 6 shows a snippet of the XSD for the sample ER diagram. The amount field from the database is wrapped in a complex type along with access levels. We utilize the standard XSD restriction tag to express the level of data available to the role. The maximum values are used to exclude data that are consumed in the report based on the user executing the report. In the current implementation, the reporting engine does not know the contents behind stored procedures and views. The stored procedures and views are added to the XSD just as tables to facilitate these objects in the business intelligence environment. In the future, we would like to interrogate the data behind these objects, so there is one tag for source data, no matter how many layers are present.

The example in Fig. 6 utilizes the maximum value a user can see based on their access level. Parent objects in the XSD are also restricted. In the case presented in Fig. 5, the patrons and campaign entities would be limited along with the donations. A parent is defined as the one side of the one-to-many relationship. In the case of patrons, it is a parent to three child entities. As long as a user has rights to one of those child entities, then the user can see the parent entity in the business intelligence system.

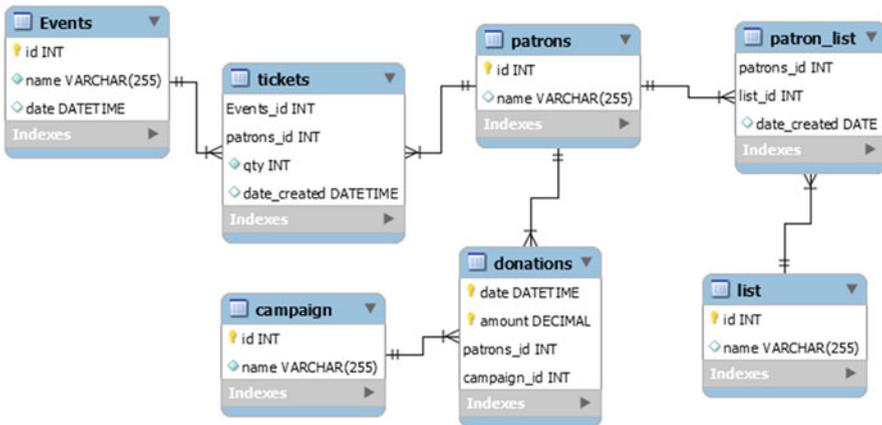


Fig. 5 Motivating example database ER diagram

```

<xs:element name="authorization_context_amount">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="amount">
        <xs:simpleType>
          <xs:restriction base="xs:decimal">
            <xs:totalDigits value="10"/>
            <xs:fractionDigits value="0"/>
            <xs:minExclusive value="-10000000000"/>
            <xs:maxExclusive value="10000000000"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="FundDevelopmentAccessLevels">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="TelemarketingStaff" >
              <xs:simpleType>
                <xs:restriction base="xs:decimal">
                  <xs:maxExclusive value="100"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="FundDevelopmentStaff" >
              <xs:simpleType>
                <xs:restriction base="xs:decimal">
                  <xs:maxExclusive value="10000"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="FundDevelopmentManagement"
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Fig. 6 XSD with privacy virtual element

The business intelligence layer will modify the query sent to the data source to ensure the filter is added to remove data where possible. In the case of SQL-based data source tables and views, the where the statement is modified to combine the filter. In the case of web services and stored procedures, the data are filtered on the middle tier (business intelligence server) before data are rendered to the client.

6 Runtime Engine

The language compiler and execution engine were built using the C Sharp programming language on the .NET Core runtime engine [11]. .NET Core is an open-source, managed execution framework that allows execution on the Microsoft Windows, Linux, and macOS operating systems. The framework is a cross-platform successor to the .NET framework. The framework allows the implementation of secBIML on any modern operating system.

secBIML links to a .NET library named Puppeteer Sharp [12]. Puppeteer Sharp is a .NET port of the Node.JS Puppeteer API [13]. Puppeteer is a Node programming language library that provides a high-level API to control the Chrome browser. Puppeteer allows a program to run the browser headless so that the browser interface is not exposed to the console. This layer of browser execution is critical in the execution of the business intelligence reports to ensure proper execution of JavaScript rendered HTML reports.

The secBIRpts server-side language is built using ASP.NET to parse the XML, XSD language documents, and to query the data sources and render the results. ADO.NET was utilized to query the data sources and to modify the queries to apply the MAC level filters.

7 Empirical Data: Data Visualization Correctness

In this section, we look at the empirical data we gathered to support our hypothesis that the usage of the secBIML language could increase the security of business intelligence reports and visualizations. To measure the availability of the business intelligence reports, we scheduled 120 reports to run overnight in six modes. The six methods were sequential with a cache and without a cache, four concurrent threads with a cache and without a cache, and eight current threads with a cache and without a cache. The tests were run over 30 days, and the average execution is shown in Table 2. Also included in the table are the average production data for the same period. The production data were gathered by parsing the web server logs for calls to the business intelligence report.

The reports that exhibit slow behavior were optimized based on the data gathered in the first phase and were optimized, and the experiment was rerun for 30 days. Table 3 shows the average timing data collected in the post-optimization period. Figure 4 shows the comparison of the average per report timing for both pre-optimization and post-optimization timing experiments. The data clearly show that the availability was increased in every mode of data gathering based on the knowledge gathered from the secBIML executions.

Table 2 secBIML pretesting data

Data point	Timing	Executions
Cache-miss sequential	17,652	120
Cache-hit sequential	1464	120
Cache-miss 4 thread	20,556	120
Cache-hit 4 thread	1824	120
Cache-miss 8 thread	22,380	120
Cache-hit 8 thread	2016	120
Cache-hit production	145,873	910
Cache-miss production	4864	320

Table 3 secBIML post-testing data

Data point	Timing	Executions
Cache-miss sequential	14,808	120
Cache-hit sequential	1452	120
Cache-miss 4 thread	18,324	120
Cache-hit 4 thread	1812	120
Cache-miss 8 thread	20,556	120
Cache-hit 8 thread	2016	120
Cache-hit production	139,647	989
Cache-miss production	4393	289

8 Empirical Data: Business Documents Correctness

In this section, we look at the empirical data we gathered to support our hypothesis that the usage of the secBIML language could increase the security of business documents. We sampled 57 business documents stored as Microsoft™ Office 365 documents. The data were stored in Word, PowerPoint, or Excel applications. Table 4 shows the documents used in our tests. The internal and external columns represent the number of criteria that we established in each category. The internal tests compare values within the document, and the external tests compare values across documents. The initial integrity column displays the percentage of the correctness of the numbers returned from the first execution of the test. The continuous integrity shows the rate of accuracy over 12 weeks. After the initial examination, corrections were applied to the documents, and continuous integrity tests ran nightly. The analysis demonstrates how often the data changed in the source data. We only found one excel document that had external budget data, and the data were correct and did not change over the 12-week test period. Discovery and setup of tests for business documents was a tedious process. In our future work, we plan to develop a Chrome web browser plugin to allow the automation of the test creation within the document. Nightly executions of the tests for business documents helped to improve the integrity, but trigger-based test execution would be a better solution. Both Microsoft Office 365 and Google GSuite offer API hooks that can be used to launch the test when a document is saved. The test could then run and immediately

Table 4 secBIML business document correctness tests

Document type	Count	Internal	External	Initial integrity	Continuous integrity
Word documents	102	24	82	82%	94%
PowerPoint documents	55	2	53	86%	92%
Excel documents	1	0	1	100%	100%

notify the user of the error. We would also plan to add web browser notifications instantly when an integrity error occurs.

9 Empirical Data: Email Correctness

In this section, we look at the empirical data we gathered to support our hypothesis that the usage of the secBIML language could increase the security of email marketing and business automation. Many CRM and email marketing vendors claim functionality to allow artificial intelligence with email marketing and continuous communication with customers based on business automation. We believe this is a more complicated process than vendors imply. The difficulty comes from the fact that the data used to generate these emails and automation must be accurate and current. So, we wanted to test the correctness of data used in a production system. To measure the integrity of the data, we used an email services provider (ESP) Mailgun [12]. An ESP is a cloud service provider that manages the delivery of email messages. Some vendors provide analytic data on email delivery, such as the number of messages delivered, suppressed, and dropped. Data about the email clients, click-throughs, and unsubscribe data are also maintained. An added benefit of the provider we chose is that a free version is available through the GitHub Student Developer Pack [13].

A Standard Query Language (SQL) Server Common Language Runtime (CLR) extension was developed to send the emails with proper tagging and retrieve the transmitted email data through the APIs. Database triggers were used to send automation responses based on the visitation of patrons. For example, an email was sent before a visitation that included details on arrival, directions to the venue, and the group's itinerary. Surveys were also sent to the patrons the day after visitation. Using the APIs from Mailgun, we were able to retrieve the data about the sent emails and check the integrity of the merged fields, appropriateness of the content in the email, and problems with delivery. Table 5 shows the errors found over a month of tests. The errors fell into two categories; data errors with the automated emails and data merge errors where data were truncated or displayed improperly in the final layout. The automation errors originated from data entry errors from operators entering transaction data and poor design in the transactional systems to allow the data inconsistencies to exist. The merge errors originated from live data that did not look like the data used in the testing of the email templates. In both cases, the

Table 5 secBIML email and automation tests

Type	Count	Errors
Visitation email automation	18,114	13
Email merge errors	756,123	1243

percentage of fault is small, but if an organization works hard to acquire a customer, these types of errors can negate that hard work.

10 Empirical Data: Privacy

The old business intelligence system in production controlled access to the data on a very coarse-grained level. Users were added to groups, and the group had access to their corresponding reports. Our preliminary research in this area replaced six reports in a test environment. The reports investigated reported on patron donations to the organization. Over these six reports, there were many data leaks that the organization did not want but tolerated because the tooling could not support a more hierarchical system. Often the data leaks were in the form of report drill downs. In a report drill down, the user is presented with a data visualization of summary data. The user is then able to click through the summary data into a display of the details. Our test implementation solved these issues but confused the users because of the missing data. We allowed aggregated data but not detail data based on the MAC level of the user running the report. Our future work needs to find a way to display full aggregate data but annotate detailed data so that missing data are still represented but not available to use to infer the real private information.

11 Conclusions and Future Work

Based on our research, we demonstrate that the availability, integrity, and correctness of business visualizations, documents, and communications increase using the secBI family of programming languages. This work demonstrates the successful implementation of the correctness tests written in secBIML for an actual organization utilizing their production environment. Our work also shows preliminary results on replacing the business intelligence platform with secBIrpts to increase privacy through the utilization of a hierarchical MAC system. Our future work will develop tooling to make it easier to create business document tests while doing layout in the document. The tooling will make it more likely that an end user will specify the correctness of a document. We will also create trigger-based executions of our testing programs. The triggers will enable on the fly verification instead of a point in time testing. We will also expand the support tags in the secBIrpts language to allow for a full replacement of the business intelligence layer. We also need to broaden the MAC privacy model to allow aggregate level data with matching drill-down data without exposing personal private details.

References

1. B. Evelson, *Topic Overview: Business Intelligence* (Forrester, 2018)
2. A. Olmsted, Secure business intelligence markup language (secBIML) for the cloud, in *Proceedings of The Eleventh International Conference on Cloud Computing, GRIDs, and Virtualization*, Nice, France, 2020
3. P. Sack, E. Austin, S. Gaetjen, Mandatory access control label security. Patent US7831570B2, 2010
4. I. Letca, A. Coelho, V.S.K. Kurapati, R. Sudhakar, D. Savage, A. Sanghvi, J. Soon Lim, Measuring actual end user performance and availability of web applications. Patent US 8,938,721 B2, 2015
5. E.F. Codd, A relational model of data for large shared data banks. *Commun. ACM* **13**(6), 377–387 (1970)
6. N. Khoury, P. Zavorsky, D. Lindskog, R. Ruhl, An analysis of black-box web application security scanners against stored SQL injection, in *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, Boston, MA, 2011
7. S. Osborn, R. Sandhu, Q. Munawer, Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Trans. Inf. Syst. Secur.* **3**(2), 85–106 (2000)
8. Oracle Corporation and/or its affiliates, MySQL, 2020. [Online]. Available: <https://www.mysql.com/>. Accessed 13 Apr 2020
9. Oracle Corporation and/or its affiliates, MySQL :: Security in MySQL :: 4.10 Using Roles, 2020. [Online]. Available: <https://dev.mysql.com/doc/mysql-security-excerpt/8.0/en/roles.html>. Accessed 13 Apr 2020
10. World Wide Web Consortium (W3C), Selectors level 3, 18 November 2018. [Online]. Available: <https://www.w3.org/TR/selectors-3/>. Accessed 16 Oct 2019
11. The Franklin Institute, The Franklin Institute, 2019. [Online]. Available: <http://www.fi.edu>. Accessed 16 Oct 2019
12. Logi Analytics, Business intelligence is dead, 2019, [Online]. Available: <https://www.logianalytics.com>. Accessed 16 Oct 2019
13. Microsoft Corporation, What is Office 365, [Online]. Available: <https://www.office.com/>. Accessed 12 Nov 2019

Framework for Monitoring the User's Behavior and Computing the User's Trust



Maryam Alruwaythi and Kendall Nygard

1 Introduction

Trust can play a key role in addressing cyber-security concerns. Cybercrimes cost the world nearly \$3 trillion in 2015. This figure is expected to increase to \$6 trillion by 2021 [1]. In recent years, the demand for computing has gradually increased. With a variety of technological advances, people can access wide-ranging information repositories and multiple resources. These interactions have become cheaper and more powerful than before. One such technology is cloud computing, which has been defined by the National Institute of Standards and Technology (NIST) [2] as “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.” There are three types of service models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Cloud computing has worthy features which are attractive to governments and business owners. By 2023, the US government will exceed \$10 billion with a growth rate of 16% from 2018 to 2023 by spending on cloud computing [3].

With cloud computing's increased popularity for providing a massive number of services, such as resources and data centers, the number of attacks is increasing [4]. The cloud platform's security is an important factor for cloud development.

M. Alruwaythi

College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia
e-mail: mruwaythi@psu.edu.sa

K. Nygard (✉)

Department of Computer Science, North Dakota State University, Fargo, ND, USA
e-mail: kendall.nygard@ndsu.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_80

1119

Storing important business data and confidential information in a cloud environment requires that a high-security mechanism be applied in the cloud platform. Basic security protections, such as firewalls and traditional access control, are not able to satisfy security requirements with the expansion of cloud computing. A lot of significant resources are stored in the cloud. The number of cloud users may be hacked because of limitations with basic security protection.

To implement a secure, reliable, and safe cloud-computing environment, we need to consider the trust issue. Therefore, it is a critical time for the cloud service provider to monitor the user behavior to detect and prevent malicious user which is the objective of this paper. This paper is organized as follows: Section 2 states the principles for evaluating user behavior. Section 3 discusses the related works. Section 4 describes the FMUBCT. Section 5 states reflecting the principles. Section 6 presents simulation results. The conclusion is in Section 7.

2 Principles for Evaluating User Behavior

In this section, we present the principles that should be considered while modeling user behavior for cloud computing [5, 6]:

- Principle 1: Expired user behavior should not be considered; when the user stopped accessing the cloud or has not accessed it recently, then the behavior records are out of date. Thus, the user should then be evaluated as a strange user.
- Principle 2: Recent user behaviors affect the trust value; new behavior must be more important and affect the trust evaluation more than long-term behavior because, with trust calculations, we consider the most recent behavior.
- Principle 3: Abnormal behavior plays an important role in trust evaluation than traditional behavior.
- Principle 4: Trust evaluation is based on a large amount of user-behavior data; the creditability of the trust value is based on a large amount of user-behavior evidence. The evidence in the cloud should be large enough to ensure that the result is stable. If the amount is small, then the results are not representative and are unstable.
- Principle 5: Slow-rise strategy is to prevent fraud risk in the trust evaluation; this strategy is based on a large number of interaction with cloud to achieve accurate trust values. This principle prevents users from gaining a high trust value when they have a small number of interactions.
- Principle 6: Punish nontrusted user based on rapid decline strategy; this strategy punishes users when abnormal behavior is detected. Punishment quickly decreases the trust value.
- Principle 7: The trust value will decrease whenever repeated malicious behaviors have occurred; repeated malicious behavior decreases the trust value more rapidly than the first occurrence.

- Principle 8: Trust evaluation should consider avoiding cheating; because the trust degree is a collaboration of different trust types, each type of trust must have weight in order to avoid too much influence being received from indirect trust.

3 Related Work

There has been research on user-behavior modeling applied to cloud environments. Bendale and Shah [7] developed a SaaS application to monitor user behavior in the private cloud; they proposed a variety of policies to evaluate the users' behavior. In the trust equation, if the user has violated a certain number of policies, the user is malicious. This model can evaluate users and can detect abnormal user behavior and malicious users when the principles are violated. However, this model does not consider the fraud risk problem and cannot prevent malicious users from receiving high trust values for short-term good behavior.

Tian et al. [5] proposed a new method based on the fuzzy AHP model to calculate the weight of behavior evidence as well as the users' trust values. In addition to improving security defenses, the authors have used multiple detection engines. These engines are used to conduct a comprehensive inspection of suspicious files. This model reflects the access time principle but fails to reflect the remaining principles. Ma and Zhang [8] proposed a new method based on improvements to the AHP method. This model considers the expiration trust record principles by creating three interaction ranges: positive, negative, and uncertain. Behavior in the negative range means that it is far from the current time and should not be included in the trust calculations. Behavior in the uncertain range means that the record is uncertain with the weight for trust calculations. Behavior in the positive range means that it is a new behavior and has a high weight for trust calculations. In addition, this model applies the principles of recent behavior and trust fraud risk through the slow-rise and punishment strategies. However, this model fails to consider the repeat abnormal behavior principle.

Xiaoxue et al. [9] proposed the reward and punishment trust model (RPTM) to calculate the users' trust values. This model is based on recommendations from other users as well as the user's historical transactions. The RPTM applies recent behavior principles and trust fraud risks through the slow-rise and punishment strategies. This model effectively differentiates between legal and malicious users. However, this model fails to consider the expiration principle and repeat abnormal behavior.

Banyal et al. [10] proposed the dynamic trust-based access control (DTBAC) model to prevent malicious users from accessing the cloud-computing environment. This model can identify malicious users and quickly prevent them from accessing the cloud server. In addition, the DTBAC has succeeded in considering the principles for the number of times accessed and the fraud risk problem. However, the DTBAC does not reflect the principles for the expiration trust record, recent behavior, and repeat abnormal behavior.

Jing et al. [11] proposed the user-behavior assessment-based dynamic access control (UBADAC) model. This model has three parts: calculating user behavior's risk values based on threat behavior; calculating user trust values based on the user behavior's risk value; and mapping user trust values, with permission. This value determines the access rights for cloud resources. This model can calculate the risk value for user behavior based on the asset value, vulnerability degree, and the threat for each resource in the cloud. The model then calculates the user's trust values based on the risk values. This model considers some evaluation principles, such as time influence and repeated abnormal behavior principles. However, the model does not consider the expiration trust record, recent behavior, or the fraud risk problem (slow-rise and punishment).

Alruwaythi et al. [12] proposed a user-behavior trust model based on fuzzy logic (UBTMFL). In this model, they developed user-history patterns and compared them with current user behavior. The outcome of the comparison is sent to a trust computation center to calculate a user trust value. This model considers three types of trust: direct, history, and comprehensive.

Based on the above analyses, this is the only (UBTMFL) model that considers all the user-behavior evaluation principles that we described in Section 2. Thus, we proposed FMUBCT to expand UBTMFL by adding additional performance indicators as user-behavior evidence. In addition, we consider an indirect trust value in comprehensive trust value.

4 The FMUBCT Model

We proposed the FMUBCT to improve the security for cloud computing by enhancing traditional access control. This improvement is achieved by checking the user's trust value before the authorized user can access the cloud. In addition, the model can monitor the user's behavior while the user interacts with the cloud in order to avoid malicious attacks from an abnormal user. The FMUBCT consists of eight primary components: the authentication, authorization, behavior monitoring, user profiling, comparison, trust computation, trust management, and fuzzy logic modules. Figure 1 illustrates the model's flowchart.

The function for each FMUBCT module is as follows:

- *Authentication Module*: verifying the user's identity when the user requests access to the cloud.
- *Authorization Module*: granting the user access to the operation and resources based on the user's trust value.
- *Behavior Monitoring Module*: monitoring of the user's behavior, in real time, throughout the interaction process. Collect the required behavior evidence; standardize the collected evidence with data preprocessing; and store the database of the user's behavior evidence.

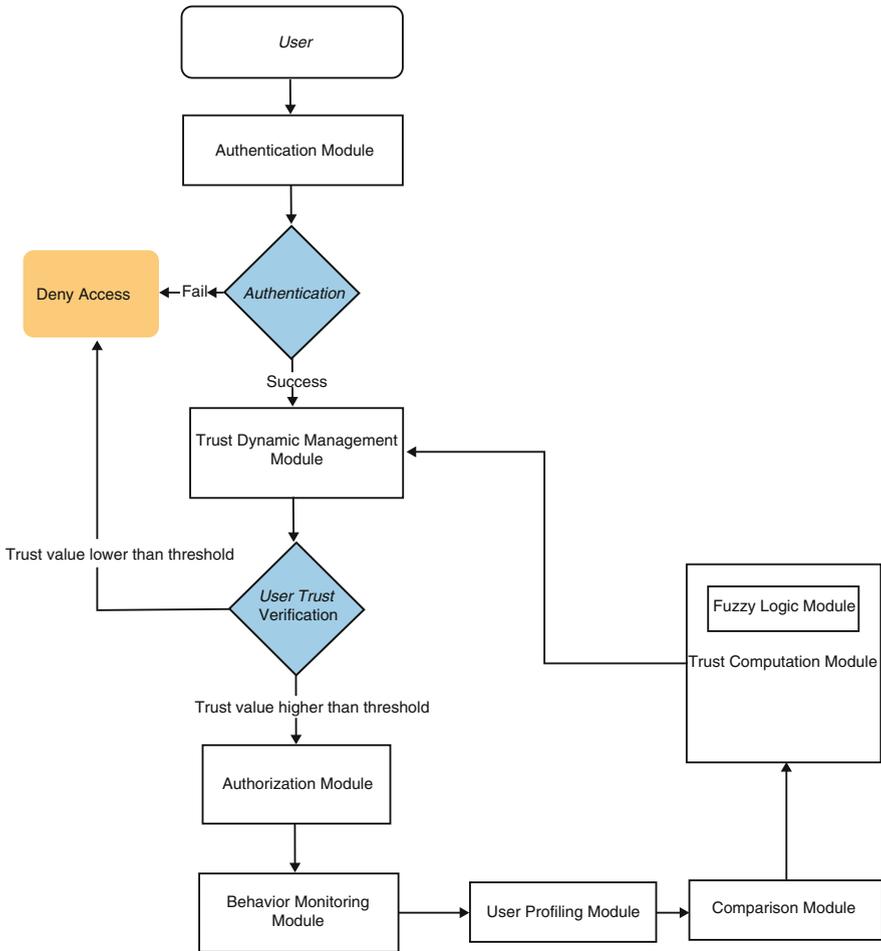


Fig. 1 Flowchart for FMUBCT model

- *User Profiling Module*: creating a user-behavior pattern based on the user’s history.
- *Comparison Module*: comparing the current user’s behavior with user’s history pattern. This comparison to obtained deviation cases, between current behavior and history pattern, are considering as malicious attacks.
- *Trust Computation Module*: applying the trust evaluation equation to the user-behavior evidence in order to calculate direct trust and history trust. Afterward, send the values to the fuzzy logic module.
- *Fuzzy Logic Module*: applying the fuzzy logic approach to compare two types of trust; then, update the user’s trust value in the trust database.

- *Trust Dynamic Management Module*: querying the user's trust value from the trust database when a user requests access to the cloud. In addition, verifying the updated user trust value dynamically modifies the user's degree of service and operating authority.

The first five modules are similar to our work, UBTMFL model [12]. In FMUBCT, we consider four evidence types, security, login, operation, and performance, to monitor the user and to collect more evidence about the user. The FMUBCT considers performance evidence because utilizing resources such as memory, disk space, and CPU are important indicators to distinguish between normal and malicious users. For example, a malicious user utilizes the entire memory by creating an extreme number of mail messages. The FMUBCT considers the following three factors:

- User's memory occupancy rate
- User's disk-space occupancy rate
- User's CPU occupancy rate

5 FMUBCT Module's Algorithms

5.1 Trust Computations Module

User behavior characteristics are various. While historical behavior data can introduce a collection of trusted behavior, user behavior does not only rely on historical behavior. Thus, using only historical data to evaluate user behavior is too indiscriminant. Therefore, we should consider different types of trust.

In this section, we describe four types of trust: direct, history, indirect, and comprehensive, which we consider in FMUBCT.

5.1.1 Direct Trust (DT)

It reflects the current interaction between the user and the cloud platform.

The model uses Eqs. 1, 2, 3, 4, 5, and 6 to calculate the direct trust.

Computation of Login Evidence

$$TL = W1 * CVLIP + W2 * CVLT, \quad (1)$$

where

- CVLIP is the comparison value of the current login IP address with the history IPs.
- CVLT is the comparison value of the current login time with the history login times.
- Weight is W , for which the total should be 1.

Computation of Security Evidence

$$TS = W1 * SIP + W2 * CV + W3 * IC + W4 * SK, \tag{2}$$

where

- SIP is the scanning important port.
- CV is the carrying virus.
- IC is the illegal connection.
- SK is the sensitive keyword.
- Weight is W , for which the total should be 1.

Computation of Operation Evidence

$$TO = W1 * CVOS + W2 * CVOA + W3 * CVOD, \tag{3}$$

where

- CVOS is the comparison value for the current service with the service history.
- CVOA is the comparison value for the current action (such as download or upload) with the action history.
- CVOD is the compassion value for the current operation’s duration with the duration history.
- The weight, W , has a total of 1. The W_i in equations 1, 2, 3, and 4 is based on the more important factor obtaining the highest value. An example of the important factor is in equation 1; when the user is scanning an important port, then SP has more weight than the other factors.

Computation of Performance Evidence

$$TP = W1 * CVC + W2 * CVM + W3 * CVD, \tag{4}$$

where

- CVC is the comparison value for the current usage of CPU with the history usage.
- CVM is the comparison value for the current usage of memory with the history usage.
- CVOD is the compassion value for the current usage of disk space with the history usage.
- The weight, W , has a total of 1.

Computation of Direct Trust Level

$$TD = \alpha * TL + \beta * TO + \gamma * TS + \delta * TP \quad (5)$$

Coefficients α , β , and γ are applied as weighting for each evidence type. The more important the evidence is to the cloud, the higher weight that it receives. The total for the weights should equal 1. To calculate the weight from equations 1, 2, 3, 4, and 5, we use Eq. 6 as follows:

$$W_i = \begin{cases} W_i = W_j & \text{where } F_i == F_j \\ W_i > W_j & \text{where } F_i > F_j \\ W_i < W_j & \text{where } F_i < F_j \end{cases} \quad (6)$$

5.1.2 History Trust

To calculate the history trust, after a long period of interactions with the cloud, we calculate the mean of the previous trust values in the positive windows. When a user logs into the cloud for the first time with normal behavior, an initial history trust value of 0.5 is assigned. The new user's history trust value is updated using fuzzy logic.

$$T_H \begin{cases} = 0.5, & \text{first login to the cloud} \\ = T_C, & \text{based on the fuzzy logic module} \\ = T_H/pn, & \text{before the fuzzy module} \end{cases} \quad (7)$$

5.1.3 Indirect Trust

Indirect trust (IT) is the score from another trusted user and same cloud provider, but from different domains; it is used to obtain the trust value for a new user in the domain when that user is an old user in another domain. In addition, if a malicious user has low trust values in other domains, then the user must be a malicious user in the new domain, too. The user's recommendation scores are stored in the database.

The total (IT) is calculated using equation 8. Any trusted user can add or update a recommendation about a user based on the experience.

$$IT = \frac{\sum_1^i IT_i}{i} \tag{8}$$

where IT is the score and *i* is the number of scores.

In this equation, we calculate the average for the recommendation scores to prevent synergies from cheating. The average protects the system from two user types: users with a smaller number of recommendations from receiving a lower score and users with a higher number of recommendations from receiving a higher score.

5.2 Fuzzy Logic Modules

Comprehensive Trust (CT) is a combination of direct, historical, and indirect trust. We use fuzzy logic to compute comprehensive trust.

We use the following methodology to calculate the comprehensive trust value of a user:

- (i) With the input as DT, HT, and IT, the output of the fuzzy system is the comprehensive trust. We illustrate fuzzification by showing the membership function for CT with a triangle view of variables (Fig. 2). We produce four membership degrees which are presented in Table 1. The equivalent membership function is shown in Fig. 2.
- (ii) Defining fuzzy rules base (Table 2).
- (iii) Evaluating fuzzy rules

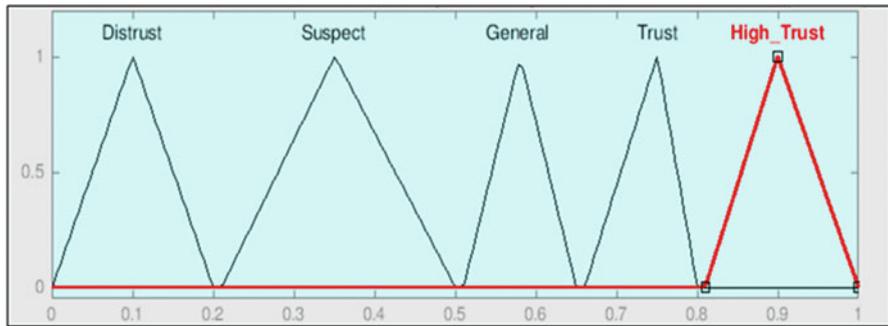


Fig. 2 Comprehensive trust membership function

Table 1 Fuzzy comprehensive trust value

Linguistic comprehensive trust value	Range	Fuzzy number
Distrust	0–0.20	(0 0.10 0.20)
Suspect	0.21–0.5	(0.21 0.3 0.5)
General	0.51–0.65	(0.51 0.58 0.65)
Trust	0.66–0.8	(0.66 0.75 0.8)
High trust	0.81–1	(0.81 0.9 1)

Table 2 Fuzzy rules

Direct trust	History trust	Indirect trust	Then
Distrust	–	–	Distrust
Suspect	Distrust	Distrust	Distrust
Suspect	Distrust/suspect	Suspect/trust	Suspect
Suspect	Suspect	Distrust/suspect	Suspect
Suspect	General	Distrust/suspect	Suspect
Suspect	General	Trust	Suspect
Suspect	Trust	Distrust/suspect	Suspect
Suspect	Trust	Trust	Suspect
General	Distrust	Distrust	Distrust
General	Distrust	Suspect/trust	Suspect
General	Suspect	Distrust/suspect	Suspect
General	Suspect	Trust	General
General	General	Distrust	Suspect
General	General	Suspect/trust	General
General	Trust	–	General
General	High trust	Distrust/suspect	General
Trust	Distrust	Distrust/suspect	Suspect
Trust	Distrust	Trust	General
Trust	Suspect	Distrust	Suspect
Trust	Suspect	Suspect	General
Trust	Suspect	Trust	Trust
Trust	General	Distrust	General
Trust	General	Suspect/trust	Trust
Trust	Trust	Distrust	General
Trust	Trust	Suspect/trust	Trust
High trust	Distrust	Distrust/suspect	Suspect
High trust	Distrust	Trust	General
High trust	Suspect	Distrust/suspect	General
High trust	Suspect	Trust	Trust
High trust	General	Distrust	General
High trust	General	Distrust/suspect	Trust
High trust	Trust	Distrust	General
High trust	Trust	Suspect	Trust
High trust	Trust	Trust	High trust
High trust	High trust	Distrust/suspect	Trust

The system must use an inference engine to obtain the fuzzy value for the output. In our model, we consider the intersection (the fuzzy AND operation) to be given as follows:

$$\text{Min } \{ \mu A (x), \mu B (x) \} \tag{9}$$

To obtain crisp output, we use the center of gravity (COG) method of defuzzification because COG is one of the most popular methods. Table 3 presents the user’s privilege based on the user’s trust value.

6 Reflecting Trust Evaluation Principles

In this section, we explain our strategies to consider the evaluation principles.

6.1 Exclude Expired Trust Value Records

When the user stops logging into the cloud for an extended period, the time between the previous trust values and the current trust value is long. We use eq. 10 to check for expiration records.

$$\text{ExR} = \text{CT} - \text{LRT} > \text{MT} \tag{10}$$

where

- CT: current time
- LRT: last record time
- MT: maximum time for the model

Because the oldest records will not be squeezed from the positive windows, we designed a strategy to exclude the expiration records. When the model detects that the user has stopped accessing the cloud for an extended time period, the model

Table 3 User’s privilege

User type	Credibility	Privilege
Distrust	0–0.3	Access denied
Suspected	0.31–0.5	Small quantity of basic services and low authority. High alert for this type of user
General trust	0.51–0.60	Basic services and general authority
Trust	0.61–0.8	Large quantity of cloud services
High trust	0.81–1	Core services and superior authority
Distrust	0–0.3	Access denied

replaces the record values with 0.5. By applying this strategy, the mean history trust is 0.5, which affects the comprehensive trust value in the fuzzy logic module. For a normal user, the trust value increases slowly by following the slow-rise strategy.

6.2 Recent User Behavior Affects the Trust Value

In the fuzzy logic module, we consider the recent behavior principle by assigning the direct trust more weight than the history trust. This principle is important to ensure that the comprehensive trust value reflects the user’s current state.

6.3 Slow-Rise Strategy

Our slow-rise strategy is that, first, we place the user in the general trust case if his/her interactions are in the uncertain windows. When the user first logs in to the cloud and behaves normally based on values from equations 1, 2, 3, 4, and 5, we replace the value with 0.51, the smallest value of direct trust, because the general trust fuzzy ranges from 0.51 to 0.65. When the user continues to behave normally in the cloud, we adjust the growth rate for the trust value by using equation 11.

$$T_D = T_D + \rho, \text{ where } \rho = (0.01, .1) \tag{11}$$

However, we retain the value from equations 6, 7, 8, and 9 if the user behaves abnormally in the cloud, meaning that $T_D < 0.5$ (Fig. 3).

$$T_D = \begin{cases} T_D, & T_D < 0.50 \\ T_D + \rho, & T_D > 0.51 \end{cases} \tag{12}$$

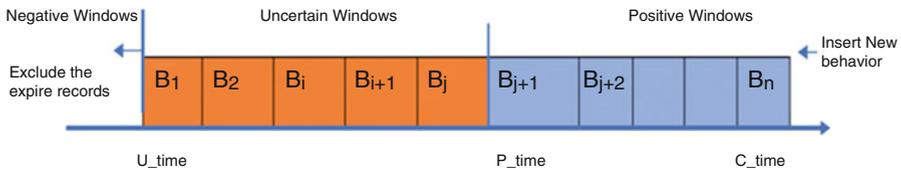


Fig. 3 Sliding windows to calculate trust

6.4 Punishment (Rapid Decrease) Strategy and Repeating Malicious Behavior

If the user behaves maliciously ($T_D \leq 0.5$), a portion of the recent trust value will be reduced to an untrusted value. Thus, the user’s T_C is rapidly decreased to punish the user.

$$\text{Punish Trust value} = \gamma * T_{\text{new}}, \tag{13}$$

where γ is the penalty coefficient used for the intensity of the punishment $\gamma = (0, 1)$.

If the user repeats malicious behavior, the γ will become lower, which decreases the final trust value by multiple penalties with the new trust value

7 Experimental Results

There is a lack of data from real system audit logs, especially in mission critical and senior industries such as healthcare, banking, and the military. Consequently, we built an algorithm-based probability theory using SAS (Statistical Analysis System) 9.4 to generate a dataset; then, we used the dataset to validate our model. Figure 4 gives an example of generating events using our algorithm.

ID	Date	Time	IP	OS	OA	OD	SP	CV	IC	SW	PC	PM	PD
1	12/12/2018	6:56	1	2	5	13	1	1	1	1	0.14	0.03	0.03
1	12/12/2018	15:58	6	8	2	5	1	1	1	1	0.05	0.12	0.33
1	12/12/2018	8:52	6	15	3	14	1	1	1	1	0.01	0.16	0.34
1	12/13/2018	15:03	6	5	1	50	1	1	1	1	0.12	0.16	0.14
1	12/13/2018	0:41	1	14	4	61	1	1	1	1	0.33	0.17	0.36
.....													
.....													
1	1/1/19	9:05	15	12	2	58	1	1	1	1	0.39	0.36	0.13
1	1/2/19	8:26	15	10	5	2	1	1	1	1	0.26	0.25	0.39
2	12/14/18	3:12	8	4	1	68	1	1	1	1	0.11	0.2	0.14
2	12/14/18	4:05	1	1	2	108	1	1	1	1	0.35	0.38	0.15
2	12/14/18	18:54	6	10	2	76	1	1	1	1	0.39	0.39	0.14
2	12/14/18	0:40	9	12	1	17	1	1	1	1	0.34	0.1	0.32
2	12/15/18	10:44	8	5	3	9	1	1	1	1	0.15	0.1	0.28

Fig. 4 Slice of generating events using our algorithm

7.1 Verification

7.1.1 Evaluate the Punishment (Rapid Decrease) and Slow-Rise Strategies

We evaluate the FMUBCT when the user behaves abnormally. Figure 5 illustrates that our proposed model can detect the malicious behavior at time 10 when the user behaves abnormally by carrying a virus to the cloud, leading to a rapidly decreased user trust value. At time 10, the FMUBCT uses equations 1, 2, 3, 4, and 5 to calculate direct trust. The result of direct trust is 0.44, and the average for the history trust is 0.68. Then, values for direct trust and history trust are sent to the fuzzy logic module to compute the comprehensive trust value. Based on fuzzy rule 8, if DT is suspect while HT and IT are trusted, then CT is suspect; therefore, the comprehensive trust value is 0.34.

7.1.2 Relationship Between Trust Types

The changing trend with three trust types is shown in Fig. 6. When the indirect trust values decrease at time 15, the history and direct trust are at the trust level, and based on the FMUBCT model, the comprehensive trust is affected. Then, at time 16, when the user behaves normally in the cloud based on the direct trust value, the user receives high repetition from other users, leading to increased indirect trust; then, the comprehensive trust increases. From this experiment, we conclude that it is important to consider more than one trust type in order to evaluate the users' behavior in the cloud. The symbols in the tables present the membership functions: N is normal; S is suspicious; G is general; D is distrust; T is trusted; and HT is High Trusted.

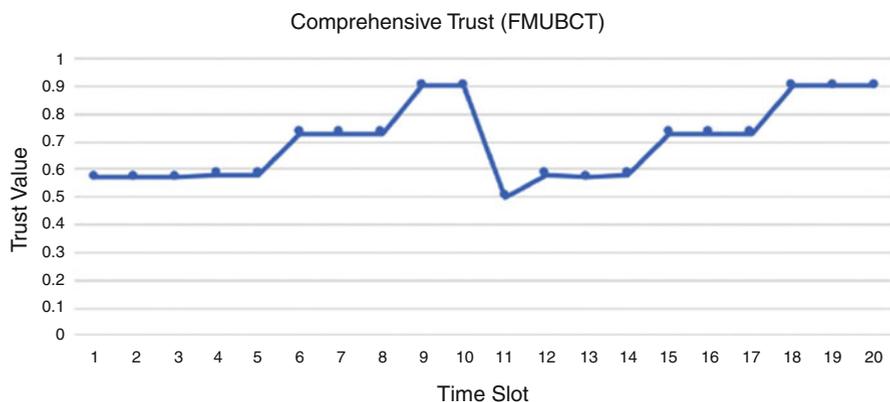


Fig. 5 Evaluating the punishment (rapid decrease) and slow-rise strategies

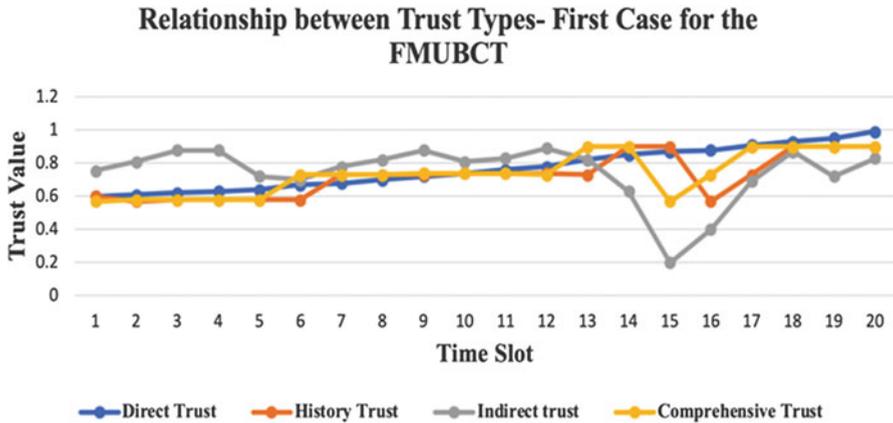


Fig. 6 Relationship between trust types

7.2 Comparative Analysis with Existing Models

7.2.1 Compute the Abnormal Detection Rate for the Models

Figure 7 shows the abnormal detection rate for the FMUBCT, the literature [7], UBADAC [11], and UBTMLF [12] models. We run 250 experiments to evaluate the effectiveness of detecting abnormal behavior. The first 50 experiments have 20 abnormal behaviors: 10 operation evidence, 5 login evidence, and 5 reliability evidence. In 100 experiments, we provide 30 abnormal behaviors: 10 operation evidence, 5 login evidence, 5 reliability evidence, and 10 security evidence. In 150 experiments, we provide 40 abnormal behaviors: 15 operation evidence, 10 login evidence, 5 reliability evidence, and 10 security evidence. In 250 experiments, we provide 60 abnormal behaviors: 15 operation evidence, 10 login evidence, 5 reliability evidence, 20 security evidence, and 10 performance evidence.

7.2.2 Comparison of the Models’ Performance

In this case, we run 3K records from dataset to compute the user’s trust value; then, we compute the mean, variance, and standard deviation. From Fig. 8, we conclude that our proposed model has a lower standard deviation than the literature model [7], UBADAC [11], and UBTMLF [12].

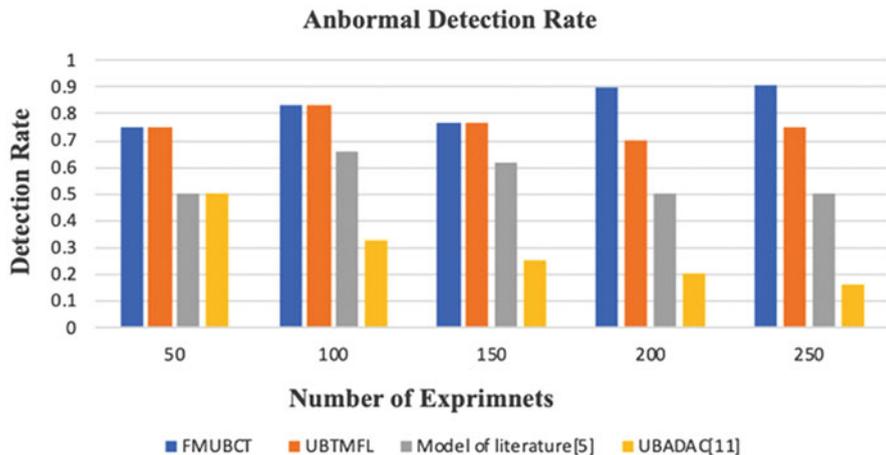


Fig. 7 Abnormal detection rate

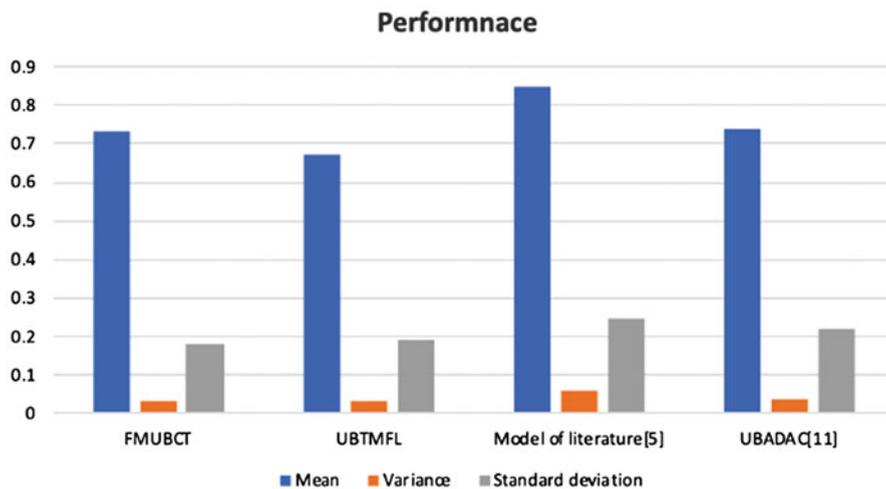


Fig. 8 Comparison of the models' performance

8 Conclusion

We proposed a user-behavior trust model based on fuzzy logic and the sliding window technique (FMUBCT) to evaluate user behavior and to detect abnormal user behavior in the cloud. This model used three types of user-behavior evidence: login, security, operation, and performance. The model has three main modules: The first module is creating a user-history pattern based on common and frequent user behaviors. Because the importance of the evidence would decrease over time,

we use the sliding window technique to describe the timelines of the evidence; then, we only consider recent user behaviors to create a user-history pattern. The second module is a comparison to contrast the current user behavior with the history pattern. The third module is trust computation to calculate user's trust. This model considers four types of trust: direct, historical, indirect, and comprehensive. We used fuzzy logic to compute the comprehensive trust value. The FMUBCT is simulated, and the results indicate that it can prevent malicious users from accessing cloud service provider whenever the user behaves abnormally; this result is achieved by monitoring user behavior and then calculating the user's trust value. Moreover, this model is able to update user's trust value in a timely manner, leading the cloud service provider to change the user's authority. Finally, the FMUBCT reflects all the identified evaluation principles, ensuring that the user's trust value is accurate. In future work, machine learning and deep learning will be computing user trust. In addition, we will use additional user-behavior evidence such as reliability. Finally, because our proposed models were built with simulated data, we can evaluate our models based on the real cloud-computing environment.

References

1. P. Paganini, Cost of cybercrime will grow from \$3 trillion (2015) to \$6 trillion by 2021, [Online]. Available: <http://securityaffairs.co/wordpress/50680/cyber-crime/global-cost-of-cybercrime.html>. Accessed 15 May 2019
2. P. Mell, T. Grance, *The NIST Definition of Cloud Computing* (NIST special publication, 2011), p. 2
3. U.S. Federal Cloud Computing Market Forecast 2018–2023, Market Research Media, [Online]. Available: www.marketresearchmedia.com/?p=145. Accessed 11 Apr 2018
4. Q. Zhang, L. Cheng, R. Boutaba, Cloud computing: State-of-the-art and research challenges. *Internet Serv. Appl.*, 7–18 (2010). <https://doi.org/10.1007/s13174-0100007-6>
5. L. Tian, C. Lin, Y. Ni, Evaluation of user behavior trust in cloud computing, in 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), (2010), pp. 567–572, <https://doi.org/10.1109/ICCASM.2010.5620636>
6. B. Dewangan, P. Shende, The sliding window method: An environment to evaluate user behavior trust in cloud technology. *Int. J. Adv. Res. Comput. Commun. Eng.* **2**(2), 1158–1162 (2013)
7. Y. Bendale, S. Shah, User level trust evaluation in cloud computing. *Int. J. Comput. Appl.* **8376**, 31–35 (2013). <https://doi.org/10.5120/12122>
8. J. Ma, Y. Zhang, Research on trusted evaluation method of user behavior based on AHP algorithm, in *2015 7th International Conference on Information Technology in Medicine and Education (ITME)*, (2015), pp. 588–592. <https://doi.org/10.1109/ITME.2015.39>
9. M. Xiaoxue, W. Zixian, B. Jing, L. Fei, Trust model based on rewards and punishment mechanism, in *2010 2nd International Workshop on Education Technology and Computer Science*, (2010), pp. 182–185. <https://doi.org/10.1109/ETCS.2010.337>
10. R. Banyal, V. Jain, P. Jain, Dynamic trust based access control framework for securing multi-cloud environment, in *Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies, ICTCS '14*, (2014), pp. 291–298. <https://doi.org/10.1145/2677855.2677884>

11. X. Jing, Z. Liu, S. Li, B. Qiao, G. Tan, A cloud-user behavior assessment based dynamic access control model. *Int. J. Syst. Assur. Eng. Manag.* **8**, 1966–1975 (2017). <https://doi.org/10.1007/s13198-015-0411-1>
12. M. Alruwaythi, K. Kambhampaty, K.E. Nygard, User behavior trust modeling in cloud security, in *The 5 Annual Conference on Computational Science & Computational Intelligence*, Las Vegas, NV, 2018

Selective Compression Method for High-Quality DaaS (Desktop as a Service) on Mobile Environments



Baikjun Choi and Sooyong Park

1 Introduction

The study suggests issues and inspection items on VDI having become the hot issue in all IT-related industries and when proceeding large-scale common use DaaS service based on this and further more seeks for solutions on these through effective gradual protocol application method [1]. The purpose of the study is to find practical approach and optimized solutions in design and realization of effective transmission for high-quality DaaS on mobile OS that should be equipped for common use and enlargement even though having any type of virtualization solutions of technologies.

1.1 Research Background

Recently wave of IT consumerization, that is, top-down decision-making structure, is changing to affect on combination of bottom-up decision-making factors. “BYOD” that performs work by using its mobile terminal is being placed as main trend. Also, to face risk of getting out of security and regulations of organization due to user’s indiscriminate use, CYOD (choose your own device) concept partly limiting rights to select terminals is appearing. BYOA (bring your own device) is a series of movement to perform work by using applications familiar to oneself, expected to reach 99% in 2020, appearing a core of consumerization. BYOA is expected to perform work providing their apps through public cloud and using private cloud for storage. At that time, for desktop function and use of application

B. Choi (✉) · S. Park
Sogang University, Seoul, Republic of Korea
e-mail: kjun@tilon.com

programs, it requires methods able to high-quality virtual program execution screen through low bandwidth in order to use on their own mobile terminals with pay-as-you-go method from VDI environment installed at large-scale IDC.

Being highly influenced by every move of Apple, worldwide platform companies holding OSs and browsers have competitively introduced mobile communication chip-based phones, tablets, and phablets and even such products that have 12-inch screens, targeting the market of the existing notebook PC. As services by which one can use multiple smart phones on a single communication network without tethering have been suggested, the explosion of demand is expected. Considering improved portability through the flexibility of display, performance beyond that of the existing PCs and communication, and worldwide commercialization of the GIGA Internet, there would be a growing demand for large-scale transmission beyond the borders of OSs or platforms.

Therefore, it is expected a demand for high-capacity computing work beyond a simple Internet search in mobile settings.

1.2 Research Contents

The study suggests issues, solutions, and optimal methods to effectively establish DaaS on 3G/LTE/Wi-Fi regardless of types of mobile OS, CPU of terminal, and manufacturer.

The main body of the study is composed of as follows. Contents and effects that should be realized for effective transmission method of high-quality DaaS are described in chapter “[Trojan Banker Simulation Utilizing Python](#).”

Results of protocol tests used in realization methods and analysis are described in chapter “[CovidLock Attack Simulation](#),” and for the last research, conclusions are addressed in chapter “[The New Office Threat: A Simulation of Watering Hole Cyber Attacks](#).”

2 Effective Transmission Methods and the Realization for High-Quality DaaS

To realize effect transmission of DaaS in mobile environment, existing compression transmission method should be actively used because generally compression methods to save network resources are related to trade-off relationship that necessarily consumes CPU resources. Such existing compression methods should be reviewed if they are able to be applied enough also in mobile devices, but since present mobile devices are equipped with enough CPU function, there are no particular problems in selecting and applying existing compression methods.

Ideas of existing compression transmission methods are able to be classified as follows:

- Individually different compression method corresponding to characteristics of content (text, pic, and movie)
- Method to transmit only screen changed (area and frame changed)

Also, for DaaS to be used on various mobile OS, following transmission methods are needed to be able to transmit screen smoothly even in low bandwidth and low specification system:

- Transmission protocol assuring QoS of high-quality contents
- Technology compressing screen fit to each property by individually separating and extracting text, image, 3D, and DirectX areas
- Adaptation type that adjusts real-time screen values changed according to network speed. Transcoding technology

A model by which the applicable technology can be actually implemented and built is suggested as follows. Also, the test environment and the entire system architecture by which the applicable method can be tested and the result values can be obtained will be mentioned (Fig. 1).

In this structure, various computing devices and smart devices located in the Internet and the intranet are separated by routers and firewalls and supplied with desktops or application programs from the server. A group of host servers which call a virtual desktop from the VM (virtual machine) image storage that stores the actual image of VM and supply it to the applicable user and the user data storage is built to save data created by users. A system which serves as a session

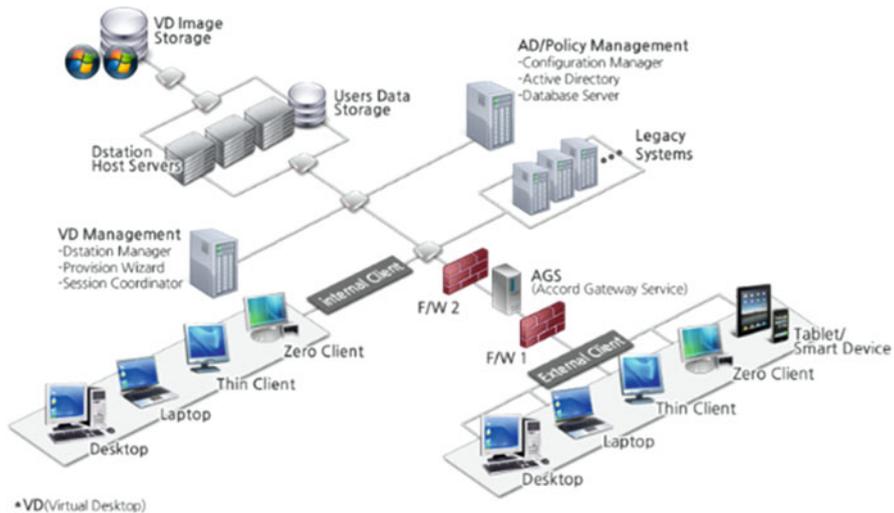


Fig. 1 A network structure comprising of DaaS server and client

broker that responds to the VM request of users and processes authorization; a provisioning tool which assigns, manages, and controls VMs to users; and the applicable VDI commercial products are installed. A gateway service is constructed which allows the connection to an internal network without setting the VPN (virtual private network) connection by the equipment through the Internet outside of the router or the mobile communication network and the smart device, and only one representative authorized IP can be opened by passing firewall. Then, for users' convenience, it can be registered at the app store to be easily installed and used.

As the abovementioned, the purpose of this study is to provide high-quality service without screen delay, flickering, or omission regardless of a user's device or OS when the user wants to receive a desktop full screen or part of application program remotely in mobile environment. The technology introduced by this study is not just to delivery contents to users but to be able to achieve the applicable purpose in the real-life situation based on security and safe certification.

2.1 System Structure

After separating text/image/movie/DirectX and 3D areas in the real-time screen area control part on DaaS server as described in Fig. 2. and selecting and applying

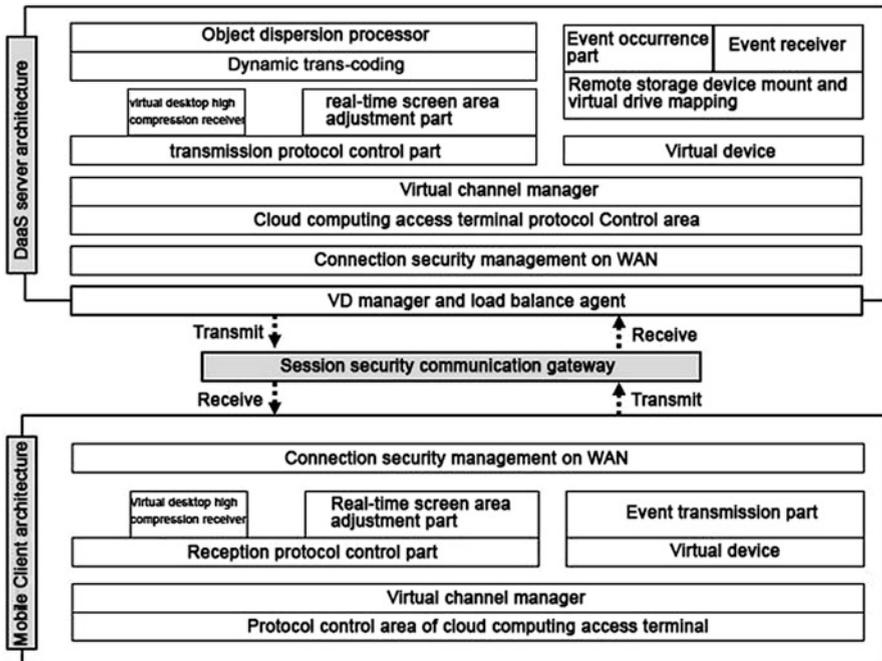


Fig. 2 DaaS server and client structure

optimal compression technology fitting to each property, dynamic transcoding is performed according to network bandwidth.

It adjusts to prevent image pushing, screen loss, and delay through event occurrence and reception in order to transmit to mobile OS and decodes by applying relevant codec to each outcome generated by optimized compression technology. After that, it visualizes effective presentation layer to users through real-time screen application.

The DaaS server object-distributed processor collects basic information (meaning user information, objective information on actual components, and the coordinate information expressed on the actual component object screen) on the server side for the object-distributed processing and information on the environment of one's hosted application program and secures basic information for object-distributed processing and environment information about the application program. Based on this, the most effective compression methods are applied performing dynamic transcoding, and preparations are ready for transmission after the compression of the object at the highly compressed transmission unit of the virtual desktop, and at the screen area control, the coordinates from the entire screen of the object are transferred to the client.

According to initialization requests, the mobile client will install an object processing engine and prepare for running of the object. The virtual application program environment that must run the object will be built based on the environment information of the application program transmitted from the event generator.

The mobile client's reception protocol controller will transmit compression output of the objects that have been transferred from the virtual desktop high concentrated receptor, and having received the relevant compression method, a decompression will be performed.

The screen area real-time controller accurately makes adjustments to accurately assign coordinates on the screen that receives the transmission; here time synchronization is controlled preventing the screen delaying and tearing phenomenon.

The virtual channel manager carries/transmits basic information for the object distribution processing and the application program environment information that has been run from the server to the client matching the information and processing an initialization request. For example, notifying of having to transmit using various types of compressed algorithms by object at the event transmitter through the virtual channel is done, and the number of each type of object, the type of relevant compressed algorithm, and together with the application method, information about the size of the relevant output and network bandwidth will be provided to the client receptor. The relevant information and compressed content that was transmitted will be matched and each decompression performed, and information about this will be sent to the server along with the ACK and virtual channel.

The screen area controller decompresses the screen area according to the individual compression algorithm displaying it on the screen, and out of each component prioritizing the component that can be immediately reflected on the screen. Meaning, in smaller data capacities first, for example, the screen is updated to accurately display the text/graphic/image/low-capacity video/high-capacity video

in the coordinates. Also, when displaying the graphic or image, the Progressive Image Rendering technique is used improving the reaction felt by users.

According to the process above the DaaS server and mobile client, after distributing the run output of application programs embedded with various types of screen layout objects, an effective algorithm is applied through compression and decompression completing the speed and bandwidth optimization process for screen layout.

2.2 *Screen Delay*

The buffering technique is used to play videos on a general live streaming service. And in the live streaming service using the compressed videos, a codec delay occurs that is required for the compression/decompression, and a propagation delay occurs due to the physical time it takes to send video data. Accordingly, screen delay is the standard the perspective takes when the user sees the final screen, and the delay occurring in the live streaming services can appear as in the following formula.

$$\text{Screen delay} = \text{Propagation delay} + \text{Codec delay} + \text{Buffering time} \quad (1)$$

Here in the propagation delay, codec delay are expressed in ms units of less than 1 second, but on the other hand, the buffering can take several seconds to several tens of seconds. This is because users viewing the streaming video want to see smooth running video regardless of the several seconds to several tens of seconds it takes to buffer. However, an interaction must take place at DaaS while looking at the screen, so reducing the screen delay becomes the priority assignment for QoS. Accordingly, buffering cannot be had like the streaming service so it becomes as follows:

$$\text{Screen delay} = \text{Propagation delay} + \text{Codec delay} \quad (2)$$

The propagation delay is very fluid according to the location of the server, or the circumstances of the circuit, but to establish a standard, we considered the long distance between Asia and the USA. The RTT between Asia (Korea, Japan, Taiwan) routers is 100 ~ 200 ms [2]. Besides, when considering the mobile environment that uses the cellular network, the propagation delay can be even greater, and even greater than that if there is screen delay from that. Accordingly, the codec delay fulfilled in the other axis must be reduced as much as possible. There is a mutual trade-off relationship with the compressed efficiency and compressed time, and it is important to find a value that optimizes the compression efficiency and compression time through appropriate experiments. It is considered that a propagation delay of 200 ~ 300 ms will occur when using the cellular network based on finding such a standard, and the goal of codec delay has been established as less than 200 ms in order to create a user's reaction speed of under 0.5 seconds.

3 Transmission Method Test and Result Analysis

Mobile OS should satisfy the following conditions to facilitate DaaS on remote IDC (Table 1).

When satisfying the condition that is able to transmit 24 frames of full HD picture in transmission time of 200 ms or under by using bandwidth of 1.5 Mbps or under, one can perform work or enjoy contents by using DaaS smoothly on smart device.

Goals for resolution quality and the play ratio of the transmitted video, network bandwidth, and the latest smart phone specs supported by FullHD were considered for installation.

3.1 Selective Compressed Codec

As the amount of data for screen update is changed according to the type of contents contained in the remote screen, efficiency of network usage and readability can be enhanced by using different compression modes. Even if the same compression method is used, the result will be different depending upon the type of contents.

Remote desktop protocol (RDP) supports multimedia redirection (MMR) for videos and H.264/AVC encoding for graphic compression as well as hardware acceleration through a virtual graphic processing unit (vGPU).

Figure 3 shows the result when the upper part and the lower part are compressed using JPEG and Zlib, respectively. Ringing artifacts is one of the representative phenomena generated in JPEG. Since it is a DCT-based compression method, it has high compression rate which is effective for moving pictures or photo transmission based on real picture while having quality degradation for text or graphic with clear boundary lines. As the UI of desktop environment is based on graphic and text, such quality degradation should be considered. Considering users' readability,

Table 1 Satisfying conditions when using DaaS

Function index	Unit	Goal
Transmission picture quality	Resolution	1080P(1920 × 1080 or 1366 × 768)
PC transmission screen quality	Frame rates	30fps or higher
Smart device transmission screen quality	Frame rates	24fps or higher
Delay time of code	Ms	200 ms or under
Use network bandwidth	Mbps	1.5 Mbps or under

Fig. 3 The ringing artifacts when an image including text is compressed



quality degradation of text or graphic should be avoided as much as possible for high-resolution and high-quality user experience.

Assume that a web browser is enlarged to the full screen on the desktop screen and compressed to be transmitted. In this case where various elements are placed complexly, it should be determined which contents are included in each of the fields and which compression method to be applied (Fig. 4).

Although the contents of a particular field are separated into a graphics field and a text field, there are some cases that text is included or mixed in graphics. For this reason, it is ideal to detect the boundaries of graphics and identify the characteristics of contents in the field detected so as to select appropriate compression method and encode the contents.

Because this separation of field could bring about delay beyond expectation, however, a VDI environment through mobile device which should be implemented in real time requires careful attention in appropriateness of field separation, as user's readability, dynamics of screen, and network bandwidth should be considered.

This experiment separated the field based on block for convenience's sake, and if graphics and text are mixed in the applicable block, the nature of field was selected according to threshold. As compressibility rate and quality are affected by threshold, the results are likely to be different in terms of policy. If the final determination could be actively selected through many experiments and quality could be determined according to network bandwidth, the desired effect could be also achieved from the service.

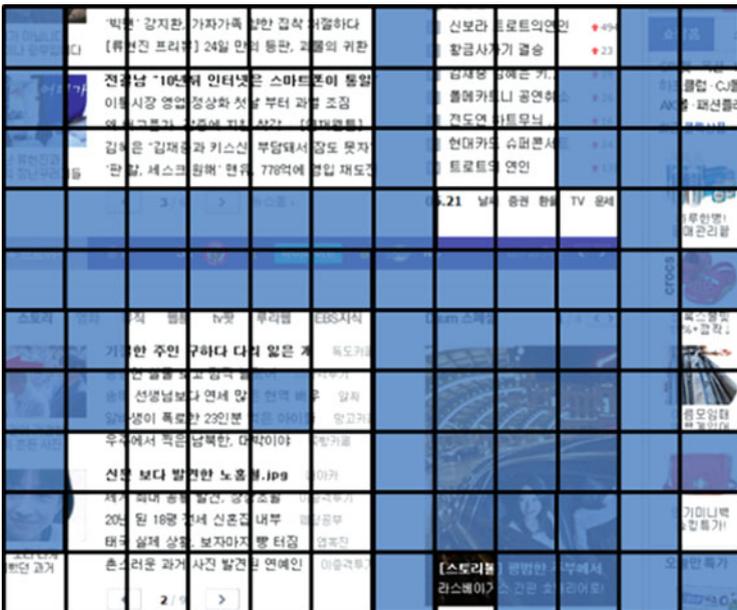


Fig. 4 An example of separation of the text field in a web page

Table 2 shows comparison on compression function according to content types, and basic environment was calculated with bites required in transmission per frame on 1024×768 resolution.

In the conclusion, HuffYUV shows lower function than JPEG or H.264 but consumes relatively less CPU resource. In the other hand, zlib compression method shows excellent function on the screen containing text, but its function drops down significantly on the screen without text. But, since compressing frames containing text with JPEG or H.264 drops down legibility of text, zlib is judged to be more effective in text-based frame transmission.

Also, in these cases of JPEG and H.264 that are available in common use, it shows evenly excellent function regardless of content.

Hence, in order to satisfying legibility of text and compression effectiveness at the same time, it is more effective to compress text containing area with zlib and compress other areas by selecting HuffYUV, JPEG, or H.264 considering CPU resource consumed.

But, since recent mobile devices contain H.264 decoding hardware to play movies, it is effective to compress with H.264 in the most cases.

Through these tests, it is discovered that selective compression method application is more effective than single compression method in HD screen transmission in order to transmit screen containing various types of contents to mobile OS devices.

3.2 Delay

Figure 5 is one part of the codec delay results when using the selective compressed codec. Several compressed methods were combined for the selective compressed codec, and there could be a difference in the compression/decompression times according to the types of contents, and such features are reflected in the experiment results. The average codec delay of the experiment was 184.63 ms, satisfying the codec delay of under 200 ms that was the initial goal.

Accordingly, the advantage that could be obtained through the selective compressed codec was that when a compression method was additionally introduced, there was the benefit of being able to replace it immediately, and also through the results, the relevant method was seen as very effective in transferring comprehensive contents through mobile devices from low bandwidth.

4 Conclusion

The study describes on methods to follow a big flow of IT consumerization era with BYOD/CYOD/BYON/BYOA and meet the demand that convenience and security of desktop are also provided in mobile environment at the same time.

Table 2 Screen compression and transmission test analysis

Category	Text	Web page	Image	Graphic	Video	3D video
HuffYUV (YV12, 12 bit)	439,948	472,504	919,748	841,692	907,784	551,280
HuffYUV (YV12, 12 bit)	936,052	1,047,216	1,852,136	1,870,600	1,884,226	1,224,461
zlib (YV12, 12 bit)	131,303	282,158	979,410	841,604	997,655	834,647
zlib (RGB, 32 bit)	191,401	525,547	2,567,978	2,230,376	2,754,009	2,448,079
JPEG (RGBA, 32 bit)	213,905	245,809	214,881	197,054	213,378	174,484
H.264	209 ~ 12,768	2231 ~ 11,175	235 ~ 9270	242 ~ 8288	354 ~ 34,131	208 ~ 24,155

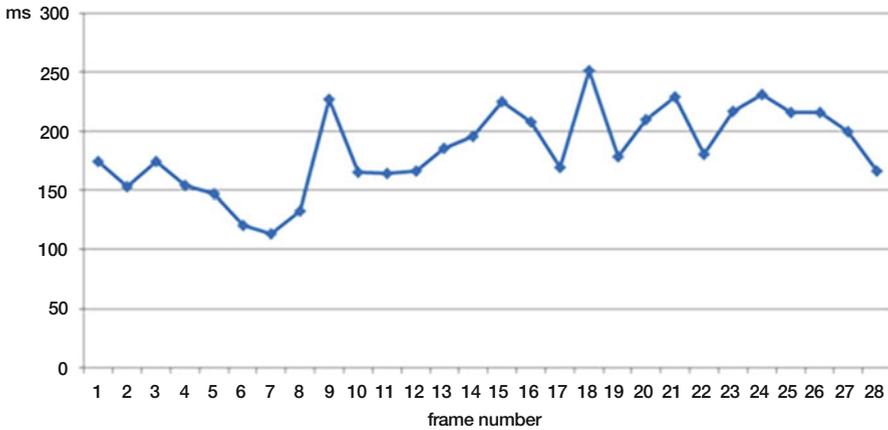


Fig. 5 Codec delay of selective compressed codec

The purpose of the study is to make DaaS capable of expressing VDI (Virtual Desktop Infrastructure) consumers so that they are charged based on the usage of mobile devices.

However, as described in the above, effective approach methods to provide smooth service in hyperconnected network environment where dozens or thousands of users are able to use at the same time were needed, but it was not easy to use low bandwidth while minimizing image loss, support, and pushing phenomenon of execution screen inside desktop.

For smooth server TM not having to depend on the bandwidth of the developed wireless network, and no matter what the place or environment of the bandwidth supported to transmit virtual machine running screen on DaaS server to remote terminal in order to solve this problem, selective compression codec application according to object and dynamic transcoding method fitting to user network bandwidth were applied and gradual rendering method and high specification graphic acceleration method were added.

Also, to be able to support more high-quality graphic application programs, object dispersion processing technology of movie component was used, which is expected to be developed to high compression virtualization technology assuring QoS of high-quality contents to future mobile OS users.

However, as entering into hyper-connected society, the conditions of user demand not only request each object's body of data not only be move, but that through such security and verification of certification, without regard to equipment or network environment, one can safely perform the smooth navigation revision/deletion/and omission actions, so it is considered that to have the appearance of a large scale cloud computing service a methodology must be added to be able to directly navigate files or a third special form of data in a virtualized environment.

The recent testing of the performance of Giga Internet showed that the download and the upload speed are 900/870 Mbps in Korea, 143/115 Mbps in Japan, 41/13 Mbps in China, and 23/3 Mbps in New York and the USA, respectively.

This indicates that the time to bring a very flexible desktop screen from the IDC in the opposite side of the earth is coming.

For mobile communication, because up to 150 Mbps of transmission speed is at hand through GiGA Path, heterogeneous network convergence technology combining LTE into GiGa-WIFI, and GiGa Wire, copper wire based high speed transmission technology, we shall encounter transmission speed 10 times faster than the current speed just in a few years.

Therefore, remote transmission of not such desktop as for business or inquiry but high-performance game or 3D screens becomes possible, and a true mobile working will be available.

Cisco expects that the number of smart devices in the world will be at least 5 billion in 2015. It means that almost the whole adults in the world have them and demands for using unlimited contents through smart devices in daily life will be increasing day by day.

For this reason, a continuous methodology for more reasonable high-quality service will be emerged for smooth transmission of remote service of large-scale desktop or application programs and contents and data held by user in the advent of hyperconnection society.

References

1. R. Buyya et al., Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Futur. Gener. Comput. Syst.* **25**(6), 599–616 (2009)
2. Global crossing, http://www.globalcrossing.com/network/network_looking_glass.aspx, (29 Apr 2014)

SURF: Optimized Data Distribution Technology



Oded Shmueli  and Itai Shaked

1 Introduction

In a hybrid cloud environment, an enterprise employs a number of local sites (or data centers) and cloud data center(s) that belong to a single or multiple cloud service providers (CSPs). SURF (security, urgency, replication, and frequency) is a technology for controlling the distribution of data and applications in a hybrid cloud environment. The data consists of *data items (dis)*. A di may have any desired granularity, e.g., an object, a file, a directory, a relation, or even a whole database. Distribution decisions are based on (a) enterprise policy, (b) performance characteristics, and (c) pricing tables. There may also be constraints, financial (e.g., on maximum expenditure) or on data placement (e.g., a specific data item must be located in Europe). Applications come in two main flavors: *one-time* that starts an operation and concludes after a certain amount of time, and *continuous* that designates systems that operate for a long stretch of time, e.g., a database server.

The first step is formulating business goals. SURF's formalism considers three main attributes of a data item: security level, usability, and replication; the SURF approach may be readily altered to accommodate additional data item attributes. For simplicity, each attribute is a natural number in the range 1 (low importance) to 10 (high importance). No generality is lost as weights are used to indicate the relative importance of the various attributes. Attribute values may be assigned to data items via a rule-based program, a SQL query, or other means.

Supported by the Israel Innovation Authority, Kamin Project 57682.

O. Shmueli (✉) · I. Shaked
Computer Science Department, Haifa, Israel
e-mail: oshmu@cs.technion.ac.il

The next step is the formulation of a quadratic integer goal program. The aim is to minimize the overall score that expresses deviation from the business goals, subject to overall constraints. Generally, constraints may express service agreements and non-linear price tables (e.g., up to N megabytes, the price is A and after that B). A goal program (GP) is a mathematical program that expresses the expected communication, computation, and storage costs of a parameterized data and application placement scenario. The solution to the GP assigns values for variables (location for data items and applications) that adhere to constraints and minimize deviation from the business goals.

Finding a solution for a GP having a large number of data items is intractable once their number exceeds a thousand or so. This necessitates treating data items in groups based on their S, U, and R attributes as well as their application affiliations. Once this is done, a solution may be obtained in reasonable time. Another facet of the SURF technology, not described herein, is *distributed algorithms* aimed at supporting data movement, while the data items are used by ongoing applications. SURF utilizes a compact ZooKeeper [2, 5] database to support these algorithms, primarily in implementing directories and lock tables.

The technology may be used as the core of the following types of system. (1) A *consultant* decision support system receives as input the current location of data items, the enterprise's policy regarding data items and applications, system characteristics (e.g., communication throughput and delays), pricing tables, and data relocation activation policy. The consultant system recommends where each data item should be stored. The system may also be used to examine various *what-if* scenarios, especially where continuous applications are concerned. (2) A *broker* is a system that has the aforementioned decision components. The broker exposes an interface similar to that of S3 [9], for storing and retrieving objects; objects are stored in one or more data centers and local sites. The broker enables object-oriented applications that utilize it for information storage and retrieval. SURF's distributed algorithms enable data movements while applications are running. The SURF platform also allows concurrency control via locking. (3) Data systems that handle their own data access, concurrency control, and replication may use a *comprehensive system* to perform actual data access. These data systems perceive a static universe although data location may seamlessly shift during their operation. (4) Finally, the comprehensive system of (3) may be embedded in a *single data system* (e.g., a relational database system or a NoSQL system) within its storage handling layer, thereby providing dynamic, optimized data placement maintenance. In system types (2), (3), and (4), user consent may be requested prior to carrying out data relocation movement.

SURF's data distribution decision technology was examined via extensive simulations on a hypothetical application and was found to be vastly superior, in a changing environment, to naive placement methods. We have also implemented a type (3) system where applications are simulated. Here too, dynamic placement decisions are superior to naive placement. Two important sub-systems will not be covered in this overview paper: data movement sub-system that reposition data while applications are ongoing and a recovery sub-system that handles failures

during data movement. It should be noted that these sub-systems are independent of the particular schemes used for concurrency control and recovery by other systems that are viewed as applications by SURF.

2 Data and Application Placement Problem Definition

2.1 Problem Description

Our goal is to optimize the placement of objects of any kind in a two-tier environment: local enterprise site(s) and data center(s) of one or more cloud operators. The placement is based upon object-level parameters, security (S), urgency of handling requests (U), and replication constraints (R), and is influenced by the measured frequency of access (F).

Conceptually, every so often the system examines the current data placement, accumulated statistics, and business environment data (such as price schemes and costs and projected data usage) and determines an optimal placement and whether it should switch to this new placement (as there is a cost associated with change). In case the decision is positive, the system implements the move while continuously providing service to its clients.

The objects (of any kind) are represented in the system by *tuples* in a small relational database, and so the optimization process finds an optimal positioning of tuples (objects), in terms of budget as well as corporate-mandated goals regarding the aforementioned SURF parameters. We make use of a goal program—a mathematical tool which allows us to formulate the various parameters of the problem in numerical form within a mathematical model and find an optimal solution to this mathematical model.

The raw output of the optimization algorithm is therefore two numbers—the *budget* expended by the system, expressed in dollars per month, and the *score* of the system which expresses deviations from the desired performance metrics, given in arbitrary units. Since the score describes deviations from the optimal desired state, *lower values are better*. The size and complexity of the goal program (in terms of number of variables and equations) is theoretically unlimited; however as the system being optimized becomes larger, the time it takes to solve the model (i.e., find the optimal solution) may get unrealistically long.

For this reason, the optimization is performed not on single tuples but rather on *groups* of tuples. Groups of tuples are stored and migrated together, and the groups themselves are defined in terms of tuple parameters (e.g., Group 1 may be defined to hold tuples with low security parameter and high replication parameter).

2.2 *Measuring and Evaluating Performance*

For the optimal solution to have any meaning, we must have a way of evaluating it against the alternative. The formulation of the goal program allows us to define constraints on the model, and so one can easily define constraints which force the model to use any data placement plan—including specifically the one currently in effect. Of course, in this case the problem becomes degenerate as the solution is predetermined, but it means the methods used to calculate the budget and score are exactly the same as the ones used to find the optimal solution, ensuring a fair evaluation.

One problem which arises from this scheme, which we have observed in early stages of constructing the optimization process, is that the group's parameter average values may not be a good-enough representation of their tuples. This is mostly due to tuples' parameters changing in between migrations, so that while at some point in the past a tuple had the correct parameter values to be in one group; by the time the optimization takes place, it may have changed. The result is that while we use the average parameter values for each group, the deviation from this average may be significant, making the evaluation of performance of the current as well as the optimized state skewed.

To alleviate this problem, each optimization run consists of three executions of the goal program:

1. The goal program is executed on a per-tuple basis, with constraints forcing each tuple to be located exactly where it currently resides. This produces a baseline score value for the system performance as well as monetary cost.
2. The goal program is executed at the group level, with no artificial constraints, producing an optimized plan for the locations of each group. The group level treats "true groups" according to tuples' *current* values for parameters.
3. The goal program is executed again on a per-tuple basis, this time with constraints forcing each tuple to be located where it would be placed under the optimal plan obtained in the previous step. This produces an accurate evaluation of system performance and price, employing the same method used in the first step.

2.3 *Choice of Tuples*

As described in the previous section, the goal program is executed twice in a degenerate setting, which is nonetheless very memory-intensive, as it represents each and every tuple in the system. Depending on the exact system being optimized, the number of tuples may grow indefinitely. This presents two problems—the first is the one of memory constraints—if tuples are constantly being added to the system, at some point the model will inevitably become too big to fit in memory. The second and arguably deeper problem is that of transient and infrequently accessed

tuples—some real-world system behaviors may produce a significant amount of tuples (data) which is created, queried, and updated over a very short time and then hardly accessed. Additionally, some tuples may be very infrequently accessed, or not accessed at all for prolonged lengths of time, then accessed multiple times over a short period. All of these cases are not handled well by the goal program, as it relies on accurate up-to-date statistics regarding the groups and the tuples—information which is not available for newly created tuples or is not indicative for ones that were not accessed for a long time.

For this reason, when running the goal program on a per-tuple basis, we do not take into account all tuples, but only those that:

1. Have been created at least 10 days prior to the time of optimization execution.
2. Have $R + \frac{HR}{2} > 4$ where R is the number of times the tuple was read this calendar month and HR is the number of times it was read the previous calendar month.

This choice inarguably introduces an error into the calculations; however the error is almost entirely in the estimation of budget, as extremely infrequently accessed tuples hardly affect system performance. The hope is that such tuples may be stored in cheap long-term storage services which incur minimal budget overhead that may be easily estimated. More work may be done in the future to assess and rectify this error if it proves to be problematic to the overall performance of the system.

2.4 *The Cost of Moving*

Of course, no change is without cost, and in our case, the cost is predominantly that of the increased communication when moving tuples or creating new replicas, collectively referred to as “moving.” The price of moving is calculated by the goal program only in the middle step, so as to weigh possibilities according to the up-front cost of executing them and not just their steady-state running cost. In order to apply this cost in a meaningful way, we need an estimation of how frequently data migrations are performed, so as to amortize the cost of moving over this period. If, for example, we only change things once a year, then a cost of 1 week’s worth of normal operation is not that bad. However, if changes are performed daily, paying such an amount for every change is unacceptable.

To do this estimation, we keep a number T_O representing the average time, in months, between executed optimizations, i.e., those implemented by moving tuples, as well as the time since the latest executed optimization plan T_L . When the time comes to calculate a new optimal plan, the goal program takes the maximum between T_O and T_L and uses it as the average time between optimization migration executions.

If the resulting plan is executed (see below), we update T_O to be

$$T_O := \frac{T_O}{2} + \frac{T_L}{2}$$

That is, the new estimate is the average between the old estimate and the time between the last executed optimization and this one. If, however, the result does not mandate a change, the value of T_O remains the same as it was. This scheme allows the estimate to settle to a relatively stable value which should reflect the actual rate of changes.

After running the goal program three times as described above, we are presented with two pairs of numbers—the current budget (meaning current expected expenditure per month) and business goal score denoted $\{B_C, S_C\}$ (obtained via execution 1), as well as the optimal budget (again, expected expenditure per month) and score $\{B_O, S_O\}$ (obtained by execution 3, using the placement outcome of execution 2). Additionally, we calculate an estimated cost for the migration M (given in dollars, as opposed to B_C and B_O which are in dollars per month). Optimally, we have managed to achieve better overall budget as well as better business goal performance, and performing the change will cost almost nothing—this, of course, rarely happens, and so we need a way of deciding when it is appropriate to execute the optimized tuple placement plan.

The following criterion is used:¹

$$\eta \frac{B_O + \frac{M}{T_O}}{B_C} + (1 - \eta) \frac{S_O}{S_C} < 1$$

where T_O is the average time in months between optimization executions and $0 \leq \eta \leq 1$ is a parameter² governing whether improving the cost or performance is more important. The first term can be thought of as the fraction of the current price that the new plan will cost—values smaller than 1 mean money will be saved, whereas values greater than 1 mean we will end up paying more. Similarly, the second term reflects the ratio of improvement in the system performance—recall that lower scores are better, and so once again any value smaller than 1 denotes an improvement, whereas greater values imply worsening of the overall performance. It is clear that as long as there is improvement in both the budget and the score, the total value will be less than 1. However, when one of the terms is greater than 1, the other must be small enough so as to make the total sum less than 1. Even so, the user may set an absolute upper bound on the cost of moving M , such that no move implementation will take place if its expected cost exceeds this bound.

¹Note that since M is in dollars and T_O in months, $\frac{M}{T_O}$ has the same units as B_O and B_C . Overall, the entire expression is unitless.

²In our simulations the value of η was chosen to be 0.7, giving more weight to budget than performance. In an operational system, the value may be user-defined.

3 Expressing Business Policies via Goal Programs

In this section we further describe the statistics and metrics required by the goal program. We also present the mathematical formulation of the goal program itself.

The goal program is a quadratically constrained quadratic programming problem (QCQP), that is, a problem of the general form:

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{subject to} && f_i(\mathbf{x}) - \mathfrak{G}_i - \mathfrak{d}_i^+ + \mathfrak{d}_i^- = 0 \end{aligned}$$

where \mathbf{x} is the vector of decision variables, f, f_i are the quadratic polynomials in \mathbf{x} , \mathfrak{G}_i are the constants, and $\mathfrak{d}_i^+, \mathfrak{d}_i^-$ are the additional *deviation variables*.³ For general background about goal programming, we refer the reader to [8] and [4].

3.1 Notation

The formulation of the goal program makes use of many different values, constants, statistics, and variables. In order to improve readability, we present an overview of our notation conventions:

Style	Usage	Examples
Bold uppercase	Sets of system components	S, G, T.
Uppercase, Greek uppercase	Constant values and statistics	$T^{i \rightarrow j}, \Delta_g^R.$
Lowercase	Indexes, decision variables, and other non-constant expressions	$x_g^j, t_g^R.$
Lowercase gothic	Deviation variables	$\mathfrak{s}_{g,i}^{\min -}, \mathfrak{u}_{g,i,j}^{W \max +}.$
Uppercase gothic	Target goal values	$\mathfrak{U}_g, \mathfrak{N}_g^\#.$
Greek lowercase	User-defined coefficients and parameters	$\alpha, \xi^i.$

Note that the only non-constant terms in the scope of a single goal program execution are **lowercase Latin letters**. Additionally, whenever applicable we have opted to place group-related indexes as subscript and site-related indexes as superscript.

Throughout the following sections, we use $H(X)$ to denote the Heaviside step function, defined

³This formulation is further explained in Sect. 3.5.

$$H(X) = \begin{cases} 1 & X > 0 \\ 0 & X \leq 0 \end{cases} \quad (1)$$

So long as X is a constant value so is $H(X)$. For decision variables we present the following general method of transforming a non-negative decision variable into a 0/1 variable, effectively implementing a step function in the framework of quadratic goal programs.

Let d be a non-negative decision variable in a quadratic goal program. Define a new 0/1 decision variable $h(d)$ along with the following constraints:

$$(1 - h(d)) \cdot d = 0 \quad (2)$$

$$1 - h(d) + d > 0 \quad (3)$$

The first constraint ensures $h(d)$ must be 1 as long as $d > 0$, while the second ensures $h(d) = 0$ in case $d = 0$. Note that the term $h(d)$ is a regular decision variable and not a function—the notation is chosen as to remind one of the meanings of this new variable—namely, the “step function of d .”

3.2 System Model

Our aim is to model a distributed system with many applications across many sites, so as to effectively quantify data storage and communication metrics (including costs). The main actors in such a system are the *sites* where data is stored and applications run; the *data*, which we think of as abstract tuples, stored on sites and consumed by applications; and the *applications*, which interact with data—reading, updating, creating, and deleting objects in response to outside operations. It is important to note that while we sometimes speak of data being sent from one site to another or a site sending data to be stored on another site, all of the operations are always performed by *applications* running on the sites and not the sites themselves. This distinction is important since a data center rarely moves, but an application may be easily moved between sites or even instantiated multiple times on distinct locations.

We start our discussion with applications, dividing the notion of an application into two parts—the *prototype* and the *instance*. An application *prototype* refers to the logic and configuration of the application—this governs the types of actions the application may perform, its read-to-write ratio, the types of tuples it may access, and so on. The same application prototype may be started or executed on many different sites—each running copy is an *instance* of that application. Instances may be executed in response to some user action (“one-time applications”), or they may be started by a system administrator and left constantly running (“continuous applications”). Where application instances can run is governed by user-defined

policies, allowing the system administrator to limit some applications to specific data centers or reserve some servers to select applications. As noted previously, only application instances perform actions in the system—every read, write, update, etc. is performed by some running instance of some application prototype. Every instance has an associated prototype, is identified by a globally unique ID, and necessarily runs on a single site.

Application instances consume and modify data. In order to avoid referring to individual data objects, we think of groups of objects, grouped together according to their characteristics, which are moved and replicated together. Since groups rely on object properties, it is reasonable to assume a high level of correlation between application prototype and the group or groups it interacts with. Groups of tuples are therefore our basic unit for both storage and communication statistics.

Finally, all data must be stored somewhere. We refer to the system component which stores data and/or hosts running application instances as a site. Note that a site may be a data center, a collection of data centers lumped together, or even a service tier inside a data center. Sites may be used to store data without running applications, may be running applications without having any local data, or may be used for both storage and applications. While the infrastructure used to store data may be distinct from that used to run applications, it may be beneficial to think of them as a single site so long as they are in the same data center and communication between them is free.

3.3 Parameters and Variables

In the following formulation, we shall make use of the following constants and parameters, divided into five categories:

1. External system description parameters, such as the number of cloud providers, their prices, and capacities. These values are taken from external sources (Table 1).
2. System directives, such as the total number of groups, SUR values, and budget limits. These values are defined by the user (Table 2).
3. Performance statistics, which are measured values regarding communication speeds, failure rates, etc. These are values that are calculated or extracted from a running system. In some cases the exact value may not be known, and so we calculate it from existing statistics (Table 3).
4. Operational statistics, which are measured values describing the system operation, such as tuple access logs. These are values that come directly from a running system, which we assume are always available (Table 4).
5. Optimization directives, including targets to be achieved by the optimizer, as well as parameters which control their relative weights. These are all user-defined parameters which allow the users to express their business goals (Table 5).

Table 1 External system parameters

Symbol	Meaning	Units
S	The set of sites. $ \mathbf{S} $ is the total number of sites	
P_s^i	Monthly storage price for site i	Cents per gigabyte
ξ^i	Security score for site i	Arbitrary, 1–10

Table 2 System directives

Symbol	Meaning	Units
G	The set of groups. $ \mathbf{G} $ is the total number of groups	
T	The set of application prototypes. $ \mathbf{T} $ is the total number of prototypes	
B	Maximum allotted budget	Cents
D_t^i	Desirability score of starting an application instance of prototype t on site i	Arbitrary
\mathfrak{S}_g	Security score target for group g	Arbitrary, 1–10
\mathfrak{U}_g	Urgency score target for group g	Arbitrary, 1–10
\mathfrak{R}_g	Replication score target for group g	Arbitrary, 1–10

Table 3 Performance statistics

Symbol	Meaning	Units
$S^{i \rightarrow j}$	Communication score from site i to site j	Arbitrary
$T^{i \rightarrow j}$	Average communication time/latency from site i to site j	Ticks per byte
P_c^i	Average communication price for site i^a	Cents per gigabyte

^aSee Sect. 3.6 for further discussion

We consider $|\mathbf{S}| \cdot |\mathbf{G}|$ 0/1 decision variables $\{x_g^i\}_{i \in \mathbf{S}, g \in \mathbf{G}}$, each denoting the replication of group g on site i . In addition to those, we also make use of $|\mathbf{S}| \cdot |\mathbf{T}|$ additional variables $\{p_t^i\}_{i \in \mathbf{S}, t \in \mathbf{T}}$ where p_t^i is the probability for an instance of type t to start on site i (given it needs to start somewhere), as well as $|\mathbf{S}|^2 \cdot |\mathbf{G}|$ additional variables $\{p_g^{i \rightarrow j}\}_{i, j \in \mathbf{S}, g \in \mathbf{G}}$, where $p_g^{i \rightarrow j}$ is the probability for an application instance running on site j choosing site i as a source for the content (data) of a group g tuple.⁴ In total the goal program consists of $|\mathbf{S}| (|\mathbf{S}| \cdot |\mathbf{G}| + |\mathbf{G}| + |\mathbf{T}|)$ decision variables.

Whenever an operation need be performed in the system which requires an application instance of prototype t to run, a choice must be made as to where (on which site) the new instance is executed. The desirability scores D_t^i are interpreted to express the relative weight of each site, that is, the probability of choosing site i as the location to run this new instance is exactly $P_i^i = \frac{D_t^i}{\sum_{j \in \mathbf{S}} D_t^j}$. This implies that whenever a policy strictly forbids the execution of some application type t on site

⁴Read: Probability of sending a group g tuple from i to j .

Table 4 Operational statistics

Symbol	Meaning	Units
N_t^i	Number of instances of application prototype t started on site i per month	
$\#R_{g,t}$	Number of read operations of group g tuples performed by instances of prototype t per month	
$\#W_{g,t}$	Number of write operations on group g tuples performed by instances of prototype t per month	
$\Delta_g^{R \rightarrow t}$	Read size of group g by t -type applications, that is, the total amount of bytes of group g requested per month by instances of prototype t	Gigabytes per month
Δ_g^R	Total read size of group g , that is, $\Delta_g^R = \sum_{t \in \mathbf{T}} \Delta_g^{R \rightarrow t}$. Note that this is only a notation shorthand, as it is completely determined by previously defined metrics	Gigabytes per month
$\Delta_g^{t \rightarrow W}$	Write size of group g by t -type applications, that is, the total amount of bytes of group g stored or updated per month by instances of prototype t . Note that this is not affected by the number of replicas	Gigabytes per month
Δ_g^W	Total write size of group g , that is, $\Delta_g^W = \sum_{t \in \mathbf{T}} \Delta_g^{t \rightarrow W}$. Note that this is only a notation shorthand, as it is completely determined by previously defined metrics	Gigabytes per month
W_g	Storage size of group g , that is, the total size of all tuples in group g^a	Gigabytes
N_g	Cardinality of group g , that is, the number of tuples in group g	
X_g^i	Matrix of current group locations— X_g^i is 1 whenever group g is currently replicated on site i and 0 otherwise	1/0
T_o	Average time (in months) between implemented optimizations	Months

^a In general we can expect $\Delta_g^R > W_g > \Delta_g^W$, as on average tuples will be read more than once per month, yet updated less than once per month, but this is by no means required

Table 5 Optimization directives

Symbol	Meaning	Units
α	Security goal coefficient, controlling the weight of Security goals relative to urgency and replication goals	Arbitrary
β	Urgency goal coefficient, controlling the weight of urgency goals relative to security and replication goals	Arbitrary
γ	Replication goal coefficient, controlling the weight of the replication goals relative to security and urgency goals	Arbitrary

i , the corresponding score must be zero. Additionally, for each application type t , there must be at least one site i such that $D_t^i > 0$, or else the application can never be started.

Once data starts being migrated, it may not be true that a site which was once a good option for running an instance of a specific prototype is still a good option, and so we must assume the mechanism controlling the instantiation sites takes data

placement into account. Explicitly, we assume that the change in the probability of choosing a site directly correlates to the portion of relevant data located on the site (i.e., applications “want” to run where most of their data resides).

To formalize this strategy, we assume the existence of two matrices $\Phi_{g,t}^R$ and $\Phi_{g,t}^W$ expressing the read and write affinities of application prototypes to the different groups. The entries of $\Phi_{g,t}^R$ (respectively, $\Phi_{g,t}^W$) are real numbers in the interval $[0, 1]$ representing the frequency of reads (respectively, writes) of group g tuples performed by t -type applications, relative to other groups. In other words, these matrices are defined as

$$\Phi_{g,t}^R := \frac{\#R_{g,t}}{\sum_h \#R_{h,t}} \qquad \Phi_{g,t}^W := \frac{\#W_{g,t}}{\sum_h \#W_{h,t}} \qquad (4)$$

For example, in a system with two prototypes and three groups, where the first prototype only accesses the first group and the second prototype accesses all groups in equal proportions, the corresponding read matrix would be

$$\Phi^R = \begin{pmatrix} 1 & 1/3 \\ 0 & 1/3 \\ 0 & 1/3 \end{pmatrix} \qquad (5)$$

Using these definitions we can define for each application prototype t a score representing how well each site’s data correlates to the application’s needs:

$$d_t^i = \sum_{g \in G} \left(\Phi_{g,t}^R \cdot \#R_{g,t} + \Phi_{g,t}^W \cdot \#W_{g,t} \right) \cdot x_g^i \qquad (6)$$

That is, for each group replicated on the site, the data correlation score is increased by a value proportional to how likely an instance of the given prototype is to access tuples of that group. This leads us to the following logical definition for the probability of choosing a specific site for running an instance of a given application prototype—the probability is a weighted sum of the *static desirability* and the *dynamic data correlation desirability*:

$$p_t^i = \begin{cases} \kappa \cdot \frac{d_t^i}{\sum_{j \in S} d_t^j} + (1 - \kappa) \frac{D_t^i}{\sum_{j \in S} D_t^j} & D_t^i > 0 \\ 0 & D_t^i = 0 \end{cases} \qquad (7)$$

where $0 \leq \kappa \leq 1$ controls the relative weight of predetermined site preferences vs dynamic data placement considerations. Since we are constructing a quadratic goal program, Eq. (7) is rearranged as a quadratic constraint:

$$p_t^i \cdot \sum_{j \in S} d_t^j = H(D_t^i) \cdot \kappa \cdot d_t^i + (1 - \kappa) \frac{D_t^i \cdot \sum_{j \in S} d_t^j}{\sum_{j \in S} D_t^j} \qquad (8)$$

Recalling $H(X)$ stands for the Heaviside step function of X , keeping with the convention of capital letters denoting constants. Note the last term need not be multiplied by $H(D_t^i)$ as it already contains D_t^i as a factor, meaning it will cancel out whenever $D_t^i = 0$.

Next we wish to formalize the probabilities of using a specific communication channel. A communication channel is simply an ordered pair of sites, denoted $i \rightarrow j$. Every such channel has various parameters influencing how desirable or undesirable it is to utilize it, which will be denoted $S^{i \rightarrow j}$ —all other factors being equal, channel $i \rightarrow j$ is $\frac{S^{i \rightarrow j}}{S^{k \rightarrow j}}$ times as likely to be used as channel $k \rightarrow j$. Since all communication is mandated by application instances (a running instance needs to read or write some data), we need only to concern ourselves with the probabilities given some data needs to be transported from or into a given site. There are two options—either an application needs to read data, in which case the data must be transported into the site running the instance, or else an application needs to write data, in which case the data inevitably originates from the site running the instance.

Given an application instance running on site j needs to read data of a group g tuple, the probability of using the channel $i \rightarrow j$ is

$$p_g^{i \rightarrow j} = \begin{cases} 0 & i \neq j \wedge x_g^j = 1 \\ x_g^i \cdot \frac{S^{i \rightarrow j}}{\sum_k x_g^k \cdot S^{k \rightarrow j}} & i \neq j \wedge x_g^j = 0 \\ x_g^i & i = j \end{cases} \quad (9)$$

The case of writes or updates is more contrived—conservatively, one might wish to always keep all replicas up to date, and so any write must be communicated to all sites holding replicas of the tuple being written. A more optimistic approach would be to only write to a majority of the replicas, or even just one, while relying on some background synchronization or other mechanisms to eventually propagate the changes to all replicas. For the rest of this document, we will assume the conservative approach is employed, i.e., updates are written to all replicas⁵

The logic behind this formulation of the probabilities is as follows. In the case of a tuple being available locally, an application will always choose the local copy and will never choose a remote copy, while in the case where a local copy does not exist, the probability of using each other site is directly proportional to its score compared to other sites where this tuple is replicated or zero if it is not replicated there.

We observe that the following conditions hold true:

$$\forall g, j \quad \sum_i p_g^{i \rightarrow j} = 1 \quad \forall j \quad (p_g^{i \rightarrow j} > 0 \Rightarrow x_g^j = 1) \quad (10)$$

⁵Less conservative approaches would necessitate sorting the probabilities in descending order and picking the first n channels.

meaning these are indeed probabilities as they sum up to 1 and that whenever there is a non-zero probability for a site to be used as the source for some group—this group consequentially must be replicated on this site. Once more, since we are constructing a quadratic goal program, we must rewrite (9) as a set of quadratic constraints:

$$\forall i \neq j, g \quad S^{i \rightarrow j} \cdot x_g^i \cdot (1 - x_g^j) = p_g^{i \rightarrow j} \sum_k S^{k \rightarrow j} \cdot x_g^k \tag{11}$$

$$\forall i, g \quad x_g^i = p_g^{i \rightarrow i} \tag{12}$$

Finally, we wish to formalize the utilization of each channel, that is, how many bytes are sent between any two given sites. For data to be transported in the channel $i \rightarrow j$, one of the following must happen:

1. An application instance of some prototype t is started on some site j , which performs an operation requiring the data of a tuple belonging to some group g replicated on site i .
2. An application instance of some prototype t is started on site i , which performs an operation resulting in the need to update data of a tuple belonging to some group g which is replicated on site j .

Therefore, the *utilization* of a channel (total number of bytes sent over the channel per month) is given by

$$u^{i \rightarrow j} = \sum_{g,t} \left(\Delta_g^{R \rightarrow t} \cdot p_t^j \cdot p_g^{i \rightarrow j} + \Delta_g^{t \rightarrow W} \cdot p_t^i \cdot x_g^j \right) \tag{13}$$

As an aside, one can think of the expression $\frac{\sum_t \Delta_g^{R \rightarrow t} \cdot p_t^j \cdot p_g^{i \rightarrow j}}{\sum_t \Delta_g^{R \rightarrow t}}$ as such: “given some data of a group g tuple is being read, what is the probability of it being read over the channel $i \rightarrow j$.” Similarly, $\sum_t p_t^i \cdot x_g^j$ can be seen as “the probability of site i writing a group g tuple into site j ,” recalling that in fact it is not the sites which perform these operations, but rather *application instances running on the sites*.

At this juncture one might wonder—why go through all this trouble to express the network utilization in terms of data locations when the same can be statically calculated through metrics? The answer is that the network utilization is highly dependent upon data placement, and so using measured (past) values implies assuming communication patterns will not change in response to data movement, which is patently false. Any change in data placement would influence future network trends in the system, and Eq. (13) attempts to capture this influence so that it can be taken into account when looking for the optimal placement plan.

3.4 Expressing Business Goals

Now we may define the monthly communication cost p_c^i as well as the monthly storage cost p_s^i for site i as

$$p_c^i = P_c^i \left[\sum_{j \neq i} u^{i \rightarrow j} + \sum_{g \in \mathbf{G}} \frac{1}{T_o} (1 - X_g^i) \cdot x_g^i \cdot W_g + \sum_{g \in \mathbf{G}} \frac{1}{T_o} (1 - x_g^i) \cdot X_g^i \cdot N_g \cdot D \right] \quad (14)$$

$$p_s^i = P_s^i \cdot \sum_g W_g \cdot x_g^i \quad (15)$$

Recalling x_g^i are the decision variables controlling the replication of group g on site i , whereas X_g^i are *constants* describing the current state of the system— X_g^i being 1 whenever group g is currently replicated on site i and 0 otherwise. The storage price is straightforward—it is simply determined by the sum of the group sizes over the proposed locations.

The communication price is made up of three terms.

The first is the expected communication cost of the normal system operation, as defined in the previous section—recall we assume local communication is always free, and so the sum is over channels between distinct sites.

The second term corresponds to the expected cost of the move itself—the storage size (W_g) of each group is multiplied by an expression which yields 1 whenever the site i is to be a new replication site for group g and 0 otherwise. Note that the usage of W_g implies that for the data migration estimate we assume the constituents of each group remain roughly the same before and after migration. In case a group changes due to optimization such that its total size is vastly different, or many of its current constituents (in terms of storage volume) are not located where the group is (according to X_g^i), this estimate will be inaccurate.

Similarly, the last term is the expected cost of deleting old copies, with D being the size in gigabytes of the message that needs to be communicated in order to delete a tuple.⁶ The term $\frac{1}{T_o}$ serves to amortize the migration cost—if data is migrated once every 2 months, the cost per month is half of the total cost, while if we change the locations every day, we will pay the transfer cost 30 times per month.

⁶If deleting a copy is considered free, D may be set to 0. If some optimization is done so that not every tuple needs a separate message, we may use an amortized value for D .

Similarly, the expected retrieval time per byte for any tuple in group g is expressed as

$$t_g^R = \sum_{i,j \in \mathbf{S}} T^{i \rightarrow j} \sum_{t \in \mathbf{T}} \frac{\Delta_g^{R \rightarrow t} \cdot p_t^j \cdot p_g^{i \rightarrow j}}{\sum_{q \in \mathbf{T}} \Delta_g^{R \rightarrow q}} \tag{16}$$

That is, for every pair of sites i, j , we take the average time per byte sent from i to j , $T^{i \rightarrow j}$, which then needs to be multiplied by the expected probability of reading a group g tuple using this channel. Tuple reads are always performed by applications, and the probability of an application of type t reading a group g tuple using the channel $i \rightarrow j$ is given by $p_t^j \cdot p_g^{i \rightarrow j}$. Since not all applications read the same amount of data of every group, this term needs to be weighted by the volume of reads of this group performed by the specific application type $\Delta_g^{R \rightarrow t}$ in proportion to the total reads of this group by all application types. Note that as per our notation, $T^{i \rightarrow j}$ is assumed to be constant or more precisely—independent of the state of the system being optimized. This essentially means we assume our choice of where to place data will never affect or saturate the inter-cloud networks—such an assumption is reasonable for all but the very largest tech companies whose data usage may comprise of a significant ratio of global internet utilization. Since the expected network latency $T^{i \rightarrow j}$ on any channel is not affected by our choices but rather only by outside influence, the goal program cannot predict it, and so it is constant as far as the solver is concerned.

In order to define goals for security, urgency, and replication, we need a way of translating the arbitrary score targets into concrete goals. We propose a method providing a mapping from the target scores into actual system variables. The actual mappings can be given as translation tables, an example of which is given in Sect. 3.7.

For the security target, we shall use two functions $\mathfrak{S}^{\min} : \{1, \dots, 10\} \mapsto \{1, \dots, 10\}$ and $\mathfrak{S}^{\#} : \{1, \dots, 10\} \mapsto \{1, \dots, |\mathbf{S}|\}$ (recall $|\mathbf{S}|$ is the total number of sites), the first signifying the minimum site security score allowed, and the second the maximum number of site replicas allowed. These specifications make sense, as a group is as secure as the least secure site that hosts it, and replicating on more sites gives more opportunities for breaches, thus decreasing security. For brevity, we shall denote $\mathfrak{S}_g^{\min} := \mathfrak{S}^{\min}(\mathfrak{S}_g)$ and $\mathfrak{S}_g^{\#} := \mathfrak{S}^{\#}(\mathfrak{S}_g)$, recalling \mathfrak{S}_g is the security score target for group g . An example is given in Table 6.

Table 6 Example security target conversion table

Security target	Min. site security (\mathfrak{S}^{\min})	Max. # of sites ($\mathfrak{S}^{\#}$)
1–3	1	10
4–6	5	4
7–9	9	2
10	10	1

Table 7 Example urgency target conversion table

Urgency target	Max. read latency (\mathcal{U}^{Rmax})	Max. write latency (\mathcal{U}^{Wmax})	Avg. latency (\mathcal{U}^{avg})
1–3	1,500ms	2,200ms	800ms
4–7	1,000ms	1,200ms	600ms
8–10	400ms	400ms	180ms

Table 8 Example replication target conversion table

Replication target	Min. # of copies ($\mathfrak{R}^\#$)	Min. % replication ($\mathfrak{R}^\%$)
1–3	1	0%
4–7	2	20%
8–10	4	50%

For the urgency target, we shall use three functions \mathcal{U}^{Rmax} , \mathcal{U}^{Wmax} , \mathcal{U}^{avg} : $\{1, \dots, 10\} \mapsto \mathbb{N}$, signifying the maximum read, maximum write, and average read transfer time in ticks. For brevity, we shall denote $\mathcal{U}_g^{Rmax} := \mathcal{U}^{Rmax}(\mathcal{U}_g)$, $\mathcal{U}_g^{Wmax} := \mathcal{U}^{Wmax}(\mathcal{U}_g)$, and $\mathcal{U}_g^{avg} := \mathcal{U}^{avg}(\mathcal{U}_g)$. An example is given in Table 7.

For the replication target, we shall use two functions $\mathfrak{R}^\#$: $\{1, \dots, 10\} \mapsto \{1, \dots, |\mathcal{S}|\}$ and $\mathfrak{R}^\%$: $\{1, \dots, 10\} \mapsto [0, 1]$, the first signifying the minimum *number* of sites on which the group should be replicated and the second the minimum *fraction* of sites (out of all sites considered) on which the group should be replicated. For example, values of 3 and 0.6 would mean a group is to be replicated on at least 3 different sites, as well as on at least 60% of the sites considered by the optimizer. Note that this means for any given number of sites $|\mathcal{S}|$ the optimizer need only consider $\max(\mathfrak{R}^\#, \mathfrak{R}^\% \cdot |\mathcal{S}|)$, and the two separate values are used mostly for user convenience. For brevity, we shall denote $\mathfrak{R}_g^\# := \max(\mathfrak{R}^\#(\mathfrak{R}_g), \mathfrak{R}^\%(\mathfrak{R}_g) \cdot |\mathcal{S}|)$. An example is given in Table 8.

3.5 Goal Program Formulation

We formulate the goal program using the method of *deviation variables*, that is, for each goal we add two artificial decision variables d^+ and d^- , called the positive and negative deviations, respectively, along with constraints: $f_i(\mathbf{x}) - \mathfrak{G}_i - d_i^+ + d_i^- = 0$, $d_i^\pm \geq 0$ where f_i is a function evaluating the i -th goal and \mathfrak{G}_i is the desired goal value. Both deviation variables are constrained to be non-negative, and so at most one of them may be positive for any given state. We use the convention that positive deviation variables signify over-achieving a goal, while negative deviation variables signify under-achieving; thus for minimum requirements, we want to minimize the negative deviation variables, as having non-zero values means the actual value was less than our desired minimum. Similarly for maximum requirements, we want to minimize the positive deviation variables.

The goal program minimizes the weighted sum of unwanted deviations (the *goodness factor*) while keeping within the allotted budget:

$$\begin{aligned}
 & \text{minimize} \\
 & \alpha \cdot \sum_g \left[\sum_i \left(s_{g,i}^{\min-} \right) + s_g^{\#\dagger} \right] + \\
 & \beta \cdot \sum_g \left[\sum_{t,i,j} \left(u_{g,t,i,j}^{R\max+} + u_{g,t,i,j}^{W\max+} \right) + u_g^{\text{avg}\dagger} \right] + \\
 & \gamma \cdot \sum_g r_g^{\#\ddagger}
 \end{aligned} \tag{17}$$

subject to

$$x_g^i \cdot \xi^i - x_g^i \cdot \Theta_g^{\min} - s_{g,i}^{\min+} + s_{g,i}^{\min-} = 0 \tag{18}$$

$$\left(\sum_i x_g^i \right) - \Theta_g^{\#} - s_g^{\#\dagger} + s_g^{\#\ddagger} = 0 \tag{19}$$

$$h(p_t^j) H(\Delta_g^{R \rightarrow t}) h(p_g^{i \rightarrow j}) \cdot \frac{T^{i \rightarrow j}}{\mathcal{U}_g^{R\max}} - 1 - u_{g,t,i,j}^{R\max+} + u_{g,t,i,j}^{R\max-} = 0 \tag{20}$$

$$h(p_t^i) H(\Delta_g^{t \rightarrow W}) x_g^j \cdot \frac{T^{i \rightarrow j}}{\mathcal{U}_g^{W\max}} - 1 - u_{g,t,i,j}^{W\max+} + u_{g,t,i,j}^{W\max-} = 0 \tag{21}$$

$$\frac{t_g^R}{\mathcal{U}_g^{\text{avg}}} - 1 - u_g^{\text{avg}\dagger} + u_g^{\text{avg}\ddagger} = 0 \tag{22}$$

$$\frac{1}{\mathfrak{R}_g^{\#}} \cdot \sum_i x_g^i - 1 - r_g^{\#\dagger} + r_g^{\#\ddagger} = 0 \tag{23}$$

$$\sum_i p_c^i + \sum_i p_s^i \leq B \tag{24}$$

$$s_{g,i}^{\min\pm}, s_g^{\#\pm}, u_{g,t,i,j}^{W\max\pm}, u_{g,t,i,j}^{R\max\pm}, u_g^{\text{avg}\pm}, r_g^{\#\pm} \geq 0 \tag{25}$$

Note that in (20), (21), (22), and (23) the target value is 1—this is because those conditions are represented as ratios, where values greater than or equal to 1 represent achieving the target.

After a minimum G has been achieved for the goal in (17), we can relax this condition by requiring the value of (17) to be at most $1.1 \cdot G$, thus turning (17) into a constraint. We proceed to minimize the left-hand side of (24)—giving the cheapest option, allowing no more than 10% deviation from the optimal corporate-mandated

goodness factor. For lack of space, we omit justifications for the various equations and possible improvements to the goal program formulation.

3.6 *Estimating Communication Cost*

In order to accurately predict communication costs as well as tuple source selection, the GP needs a way to predict the price per byte sent out of every site (we assume inbound traffic incurs no cost). Naively, one could take for each site i the total bytes sent out of i from the start of the month to the current point in time and divide it by the cost of communication for this site for the same time frame. This approach fails in giving accurate predictions, though as real-world cloud providers almost exclusively employ a non-linear pricing scheme, where the first few GB/month are free of charge, with prices per subsequent usage decreasing as the total amount increases (Fig. 1). Such pricing schemes will make the naive approach grossly underestimate the cost in cases where most of the communication to date did not incur any cost—either because the optimization was run early in the month or because the specific site had little or no data stored on it (e.g., a new service is introduced).

A somewhat more balanced approach, and the one the current optimizer uses, assumes the overall communication needs of the system remain constant and estimates the relative price bracket assuming a relatively even distribution. In practice, we keep two total counters—the total outbound communication size in the previous calendar month, denoted Δ_H , as well as the same metric for the current calendar month,⁷ denoted Δ . At any given moment, assuming $0 < M < 1$ denotes how far along the current calendar month we are (0 being the very first moment in the month, 0.5 being exactly halfway, etc.), we estimate the total system outbound communication for the entirety of the current calendar month to be

$$\Delta_{\text{est}} = (1 - M) \cdot \Delta_H + \Delta$$

Note that Δ need not be weighted, as it is already weighted by the fact that we only have information for the M th part of the month. We then proceed to assume the communication cost bracket for each site, derived from the total outbound communication originating from this site, to be roughly determined by the average of this value $\frac{\Delta_{\text{est}}}{S}$ (recall S is the total number of sites to be considered). While assuming a uniform distribution is patently incorrect, it should at least give us a ballpark estimate for where we expect each site to end up in terms of communication cost per byte. The main advantage of this approach is that the calculation is done

⁷The reason for using a gross total, as opposed to a running window spanning exactly one month, is that of simplicity—it is much simpler to store only two values than it is to keep track of all historic transactions.

prior to running the GP, and so as far as the GP is concerned, all values discussed are predetermined coefficients (P_c^i).

3.6.1 Improving the Estimate

If the previously discussed method is deemed inaccurate, another possibility is to introduce additional variables into the GP, which scale the effective cost in accordance with the configured communication price curve. Recall we assume a pricing model based on quotas, each with a cost rate given in cents per gigabyte. Denote by K^i the number of such steps for site i , labeled $1, \dots, K^i$, by Q_k^i the k -th quota (in gigabytes) for site i and by R_k^i the cost rate starting from the k -th quota.⁸ As a convention, we define $R_0^i = 0$ and $Q_{K^i+1}^i = \infty$.⁹

As an example, consider the communication cost graph depicted in Fig. 1, and suppose we have sent 20 TB (i.e., 20,000 GB) of data out of the site in question. Figure 2 shows the labels for the step quotas as well as rates, with the shaded area representing the total cost of communication out of this site (for clarity we have omitted the site indexes in this example). Note that the first gigabyte of data sent out incurs no cost, as $Q_1 = 1$ in this example—if there is no free tier, the value of Q_1 would be zero, meaning you pay right from the first byte.

To get an accurate formulation of these cost steps, we first look back at (14), extracting only the communication *volume* out of site i as

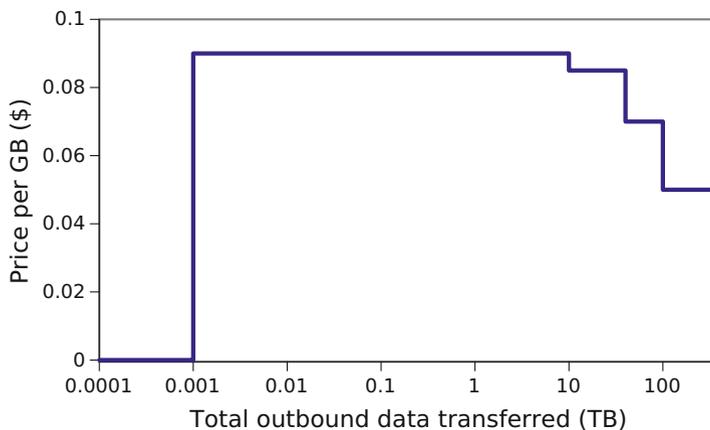


Fig. 1 Characteristic communication price graph

⁸That is, you pay R_k^i cents for each gigabyte of communication out of site i from Q_k^i to Q_{k+1}^i gigabytes.

⁹That is, egress communication up to the first quota is free of charge, and the last step is valid for arbitrarily large values.

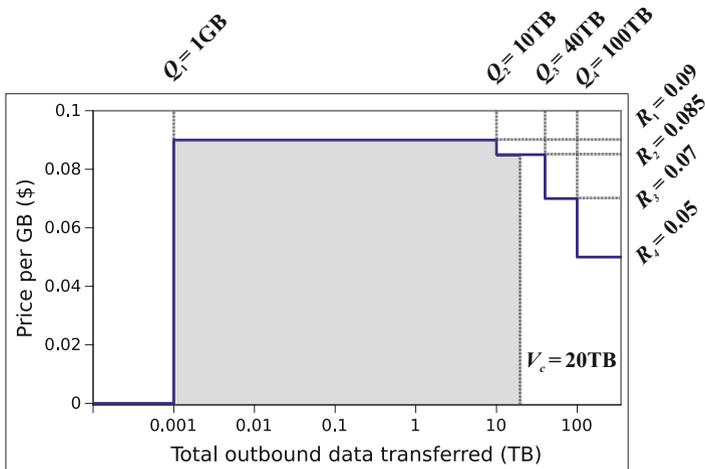


Fig. 2 Example of communication cost calculation

$$v_c^i = \left[\sum_{j \neq i} u^{i \rightarrow j} + \frac{1}{T_o} (1 - X_g^i) \cdot x_g^i \cdot W_g + \frac{1}{T_o} (1 - x_g^i) \cdot X_g^i \cdot N_g \cdot D \right]$$

Conceptually, the exact communication cost for site i is the sum

$$\begin{aligned} &R_1^i \cdot (\text{communication volume from } Q_1^i \text{ to } Q_2^i) \\ &+ R_2^i \cdot (\text{communication volume from } Q_2^i \text{ to } Q_3^i) \\ &\dots \\ &+ R_{K^i}^i \cdot (\text{communication volume from } Q_{K^i}^i \text{ to } Q_{K^i+1}^i) \end{aligned}$$

So we need to find a way to express this sum using the framework of a quadratic program.

To do so, for each site i , we define K^i new deviation variables, $\{\mathfrak{V}_k^{i\pm}\}_{k=1 \dots K^i}$, along with the appropriate constraints:

$$v_c^i - Q_k^i - v_k^{i+} + v_k^{i-} = 0$$

such that the value of v_k^{i+} is the amount of gigabytes *over* the k -th quota step, including any higher step. It follows that $(v_k^{i+} - v_{k+1}^{i+})$ is exactly the amount of data to be billed according to the rate of the k -th step. Finally, we may replace (14) by:¹⁰

$$p_c^i = \left[\sum_{k=1}^{K^i} R_k^i (v_k^{i+} - v_{k+1}^{i+}) \right] \tag{14'}$$

Going back to our example in Fig. 2, this formulation results in the following calculations (site indexes omitted again, for clarity):

$$\begin{aligned} v_c &= 20,000 \\ 20,000 - 1 - v_1^+ + v_1^- &= 0 & \Rightarrow v_1^+ &= 19,999 \\ 20,000 - 10,000 - v_2^+ + v_2^- &= 0 & \Rightarrow v_2^+ &= 10,000 \\ 20,000 - 40,000 - v_3^+ + v_3^- &= 0 & \Rightarrow v_3^+ &= 0 \\ 20,000 - 100,000 - v_4^+ + v_4^- &= 0 & \Rightarrow v_4^+ &= 0 \end{aligned}$$

and so the total cost would be

$$\begin{aligned} p_c &= R_1 (v_1^+ - v_2^+) + R_2 (v_2^+ - v_3^+) + \\ & R_3 (v_3^+ - v_4^+) + R_4 (v_4^+ - v_5^+) \\ &= 0.09 (19,999 - 10,000) + 0.085 (10,000 - 0) + \\ & 0.07 (0 - 0) + 0.05 (0 - 0) \\ &= 0.09 \cdot 9,999 + 0.085 \cdot 10,000 \end{aligned}$$

That is to say, out of the total 20,000 gigabytes, the first gigabyte was free of charge, the next 9,999 (up to 10TB, but not including the first GB) were billed according to the first rate of 9 cents per gigabyte, and the last 10,000 (from 10TB up to our total usage of 20TB) were billed according to the second rate of 0.085 cents per gigabyte. Since we have not surpassed the third quota of 40TB, no data was billed according to the last two rates.

While at first (14') may seem much simpler than (14), recall that each v_k^{i+} is defined in terms of the volume v_c^i , so (14') is essentially (14) with multiple coefficients $\{R_k^i\}_{k=1}^{K^i}$ instead of the previous P_c^i . This approach, while being

¹⁰Note our convention that $Q_{K^i+1}^i = \infty$ implies $s_{K^i+1}^+ \equiv 0$.

inarguably more accurate, would increase the number of variables in the GP by $2 \cdot \sum_i K^i$. This in turn may negatively impact performance (optimization time).

3.7 Example Target Translation Tables

For clarity, we include examples of target goal translation tables. The tables show one way of providing goal value translations for the various group target parameters.

We outline three main ways to optimize both applications and data placement:

1. With each data item, indicate which continuous applications use it. Add a *constraint* that states that for each continuous application S there is at least one site (data center or availability zone) which holds all data associated with application S . Then, make placement decisions for all data, and the placement indicates in which sites each continuous application should be placed. In operation, *dynamically* place one-time applications at the best places, based on their data requirements while, probabilistically, balancing such placements.
2. Place continuous applications once (user-directed, not a SURF decision). Thereafter, make placement decisions for data only and *dynamically* place one-time applications at the best places while balancing the computing load. This scheme fits many commercial settings. The *once* part may be repeated at certain (relatively long) time intervals. Continuous applications are categorized as *single* or *multi*, namely, restricted to a single installation (single), or can there be more than one installation running them (multi).
3. All *applications and data items* are placed. The continuous ones are as in the previous methods. The one-time applications are subdivided into *classes* where each class is similar to a continuous application (although each of its constituents is a one-time application). Each class member is associated with a specific collection of data items (usually specified by ranges) and the expected number of such application instances. Data items are associated with such classes as in the second method. At run time, actual instances are executed as in the previous cases (load balancing, best places).

4 Data Placement Experiments via Simulations

4.1 Experimental System

we constructed a simulation system, allowing us to test and re-test long-term scenarios in relatively short periods of time.¹¹ Every scenario was run through

¹¹A typical scenario simulating the evolution of a system over a 6-month period would play out in 12–24 h.

the simulator two times employing the same random seed, where one run was configured to periodically run the optimization algorithm, moving data if deemed beneficial, while the other was allowed to perform two optimizations, after which no data migration was performed. The reason for performing two optimizations on the static system was to allow a fair starting point. For example, a scenario was run for 6 (simulated) months, with optimization evaluations every 5 days. 7 sites were considered—one local site and 6 cloud sites arranged as two distinct data centers with three performance tiers each. Eighty-one groups were defined by splitting each of S, U, R, and F into three ranges (1–3, 4–7, and 8–10). Fifteen thousand five hundred initial data tuples were used, totaling $4.025 \cdot 10^9$ bytes. Every 5 days 1.5% of tuples' parameters were randomly chosen and changed to simulate changes in data characteristics and requirements. The results show that while both systems start out having equivalent performance, the static system's performance quickly deteriorates (Fig. 3). More importantly, the optimized system maintains a stable level of performance *without increasing the total cost*, even though it has the additional cost of moving data between sites (Fig. 4). This behavior is characteristic

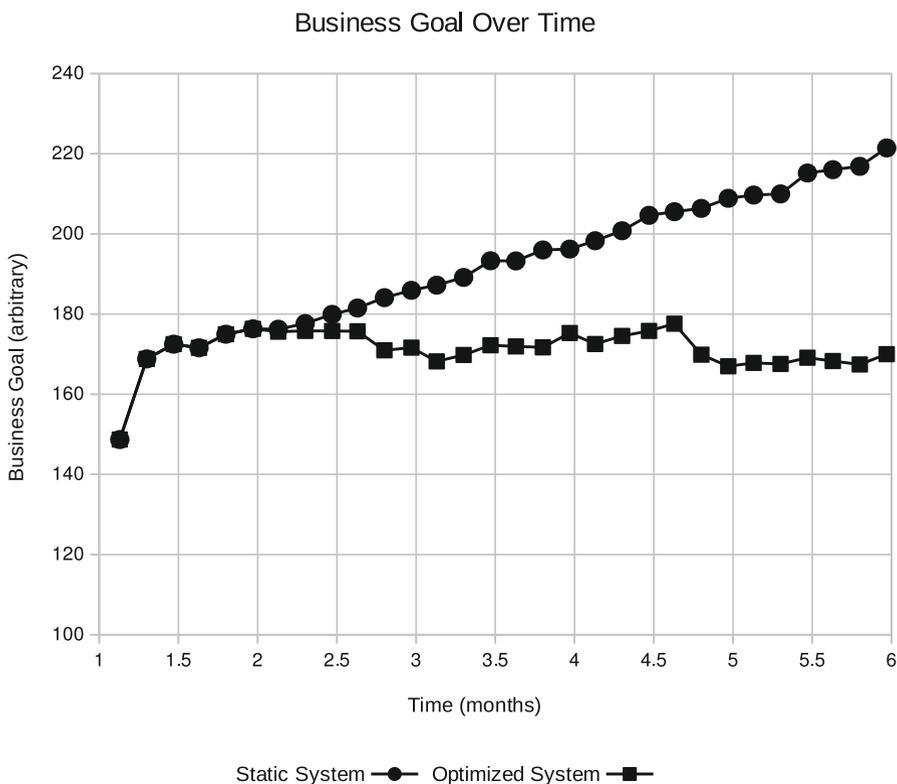


Fig. 3 Business goal comparison for scenario A. The performance of the static system degrades over time, while the optimized system maintains a relatively stable level of performance

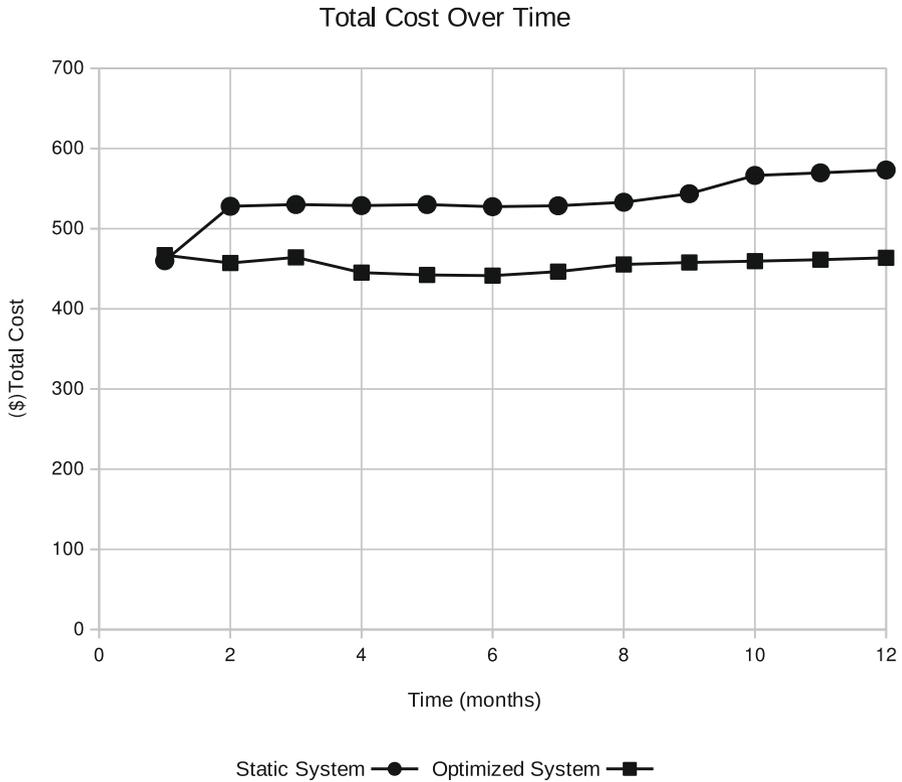


Fig. 4 Total cost comparison for scenario A. The optimized system managed to achieve better performance without increasing total cost. By the end of the scenario, the optimized system even shows a slight saving in total cost

of scenarios where tuples' parameters change over time—a placement that was once optimal might not remain optimal as time goes by, with the result of paying for performance where it is not needed while not allotting resources for tuples that do require it.

5 Related Work

There is no notion of business goal per object in the works we surveyed. For example, SPANStore [11] optimizes object placement based on price discrepancies while meeting latency constraints. Computation is done at a *single* provider's data centers, and storage is at various providers. [6, 7] consider dynamic migration and replication in view of price differentials, storage classes, and usage; its goal is minimizing monetary cost. Unlike our work, The work of [10] considers a scenario in which

an application serves various geographic domains on a *multi-cloud*. Throughout the day, the intensity decreases in certain geographic domains and increases in others. This motivates *virtual machine migration*. Similarly to SURF, the system migrates data and virtual machines as a unit based on optimization and seems especially suitable to cyclical changing global geographic loads. Hajjat et al. consider an enterprise system containing servers running *components* [3]. Components are either front-end, business-logic, or back-end components. A component may be run on more than one server, each supplying the same functionality. System users are classified as internal and external. Components may be storage oriented, e.g. back-end databases, or computing oriented. The objective is to migrate some servers running components to the cloud in a way that saves money and adheres to enterprise policies and, in particular, to communication delays mean and variance constraints. Similar to our work, communication delays are central. However, these delays have to do with servers and components and not with data items. The work of [1] presents a system for cost optimization placement of applications.

6 Conclusions

We presented the SURF technology for data and application placement decisions and actual data movement. Data placement is a complex problem. On the one hand, placement needs to adhere to the enterprise's policy (expressed via business goals) related to the level of security of the data, required quickness of data access, required replication level, data access statistics (frequency), and a myriad of performance parameters (such as capacity and communication characteristics and economic considerations such as budget cap and pricing tables of various providers).

Business goals are expressed using the formalism of *goal programs*. The modeling expressed in the resulting goal program (GP) is non-trivial and faithful to actual systems. Given the multitude of parameters and the number of data items, we presented an efficient method (grouping) for making the solution tractable by an industrial strength integer-linear solver. We have implemented a consultant system based on this technology. SURF's data distribution decision technology was examined via extensive simulations on a hypothetical application and was found to be vastly superior, in a changing environment, to naive placement methods.

References

1. O. Belli, C. Loomis, N. Abdennadher, Towards a cost-optimized cloud application placement tool, in *2016 IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2016*, Luxembourg, December 12–15, 2016 (IEEE Computer Society, 2016), pp. 43–50. <https://doi.org/10.1109/CloudCom.2016.0022>
2. Foundation, A.S.: Apache zookeeper (2019). <https://zookeeper.apache.org/>. Online Document

3. M.Y. Hajjat, X. Sun, Y.E. Sung, D.A. S.G. Maltz, Rao, K. Sripanidkulchai, M. Tawarmalani, Cloudward bound: planning for beneficial migration of enterprise applications to the cloud, in *Proceedings of the ACM SIGCOMM 2010 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ed. by S. Kalyanaraman, V.N. Padmanabhan, K.K. Ramakrishnan, R. Shorey, G.M. Voelker, New Delhi, India, August 30–September 3, 2010 (ACM, 2010), pp. 243–254. <http://doi.acm.org/10.1145/1851182.1851212>
4. J.P. Ignizio, *Goal Programming and Extensions* (Lexington Books, 1979)
5. F.P. Junqueira, B.C. Reed, M. Serafini, High-performance broadcast for primary-backup systems, in *Proceedings of the 2011 IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2011*, Hong Kong, China, June 27–30 2011 (IEEE Compute Society, 2011), pp. 245–256. <https://doi.org/10.1109/DSN.2011.5958223>
6. Y. Mansouri, R. Buyya, To move or not to move: Cost optimization in a dual cloud-based storage architecture. *J. Netw. Comput. Appl.* **75**, 223–235 (2016). <https://doi.org/10.1016/j.jnca.2016.08.029>
7. Y. Mansouri, A.N. Toosi, R. Buyya, Cost optimization for dynamic replication and migration of data in cloud data centers. *IEEE Trans. Cloud Comput.* **7**(3), 705–718 (2019). <https://doi.org/10.1109/TCC.2017.2659728>
8. C. Romero, *Handbook of Critical Issues in Goal Programming* (Pergamon Press, 1991)
9. A.W. Services, Amazon s3 (2020). <https://aws.amazon.com/s3/>. Online Document
10. Z. Shen, Q. Jia, G. Sela, B. Rainero, W. Song, R. van Renesse, H. Weatherspoon, Follow the sun through the clouds: application migration for geographically shifting workloads, in *Proceedings of the Seventh ACM Symposium on Cloud Computing*, ed. by M.K. Aguilera, B. Cooper, Y. Diao, Santa Clara, CA, USA, October 5–7, 2016 (ACM, 2016), pp. 141–154. <http://doi.acm.org/10.1145/2987550.2987561>
11. Z. Wu, M. Butkiewicz, D. Perkins, E. Katz-Bassett, H.V. Madhyastha, *SPANStore*: cost-effective geo-replicated storage spanning multiple cloud services, in *ACM SIGOPS 24th Symposium on Operating Systems Principles, SOSP '13*, ed. by M. Kaminsky, M. Dahlin, Farmington, PA, USA, November 3–6, 2013 (ACM, 2013), pp. 292–308. <https://doi.org/10.1145/2517349.2522730>

Securing Mobile Cloud Computing Using Encrypted Biometric Authentication



Iehab AlRassan

1 Introduction

Cloud computing is a hot topic nowadays because of the growing spread of mobile users and the simplicity and functionality of mobile features. MCC allows all information in the virtual environment to be shared by allowing users utilizing software, platforms, and infrastructure environments in real time, as they need. Therefore, MCC is the future of computing systems that enables the control, management, and retrieval of data [1]. Thus, the user, as well as the service, must be authenticated to ensure the privacy and the security of the cloud service that is being provided, and this is considered a major challenge in cloud computing. A major challenge in both cloud and mobile cloud computing is ensuring the privacy of each user's personal information.

Mobile cloud computing (MCC) is known as a service of cloud computing which is offered in either a mobile phone environment or an environment with a mobile embedded system. It has been created through the integration of mobile computing with cloud computing. MCC is a type of infrastructure where storage and processing of data are performed outside a mobile device. Mobile cloud computing has inherited the main characteristics of the cloud model: broad network access, on-demand self-service, resource pooling, rapid elasticity, and measured services [2].

Recently, mobile phone usage has been increasing, and most people have been using their phones to access the internet, their e-mail, etc. With this growth, the need for security has increased in order to satisfy protection requirements. Authorized

I. AlRassan (✉)

College of Computer and Information Sciences, King Saud University, Riyadh, Kingdom of Saudi Arabia

e-mail: irassan@ksu.edu.sa

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Parallel & Distributed Processing, and Applications*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-69984-0_83

1177

access and biometric authentication fields as well as encrypted data can be saved to the cloud to protect data against third party or cloud provider. Although there are two layers of authentication, which are user password and biometric authentication (fingerprint), the data might still be exposed to an unauthorized user or a third party (provider). Because of this risk, data in the cloud will be encrypted using user fingerprint and password to generate secured channel for exchanging user key. However, the question is how to achieve efficiency and privacy when encrypting this data. Traditional encryption schemes can be used for protecting data in cloud computing, but this data cannot be used for any computation without the private key to decrypt it, resulting in the data being exposed and the privacy being violated which is against the main contribution.

Homomorphic encryption can solve this problem; it allows the cloud to perform any computation without the need to decrypt it and without any knowledge of the private key. Only the user has access to the private key, and this guarantees the security of the encrypted data.

This research will significantly contribute to the enhancement of these sectors so that mobile cloud computing accounts can be more accurate and better protected. Moreover, protecting user's identity and data will protect users' privacy and prevent third parties and attackers from tracking users' or knowing users' activity.

The remaining of this paper is organized as follows: Section 2 discusses the literature review of security in mobile cloud computing and encryption techniques; Section 3 discusses the proposed system in details; Section 4 explains discussion and results; and, finally, Section 5 addresses the summary and conclusion.

2 Literature Review

In [3], the authors defined cloud computing as a model for enabling a convenient and on-demand network that makes use of a shared pool of configurable computing resources (such as networks, servers, storage devices, applications, services, etc.).

In [1, 2, 4] and [5], they explained the three different layers that cloud computing can incorporate to provide services; they are software as a service (SAAS), platform as a service (PAAS), and infrastructure as a service (IAAS). Cloud computing can differ depending on requirements specified for specific characteristics that support the needs of services and users of clouds. These deployment models are available on the private cloud, community cloud, public cloud, and hybrid cloud.

Because of the nature of cloud as a sharing resource, the data and users are more vulnerable to a security breach. In [6], they classified the security issues in MCC in relation to mobile users and data security threats. They described three security factors that affect the data stored in the cloud such as integrity, which is an essential feature of any security framework that ensures data consistency and protects it from unintended alterations. A user's privacy, location, and other private information provided by the user might be accessed and misused by an attacker if it is not carefully protected. Authentication is the process of confirming the identity

of a user who attempts to access a resource or service. The authentication process is important in the mobile cloud computing; the authors mentioned a number of different authentication mechanisms either traditional or novel.

In [7], it focusses heavily on security issues, access control, and authentication as the main concerns. They proposed in their paper many authentication methods that provide their strengths and weakness.

In [8], the authors of this paper conducted a survey on the main motivations of mobile cloud computing summarizing its challenges, and they presented a taxonomy of issues found in this area. They focused on operational-level issues, end user-level issues, service- and application-level issues, privacy, security and trust, context awareness, and data management.

In [9], the paper discussed mobile cloud computing, which combines the advantages of both mobile computing and cloud computing that provides the optimal services for mobile users. This paper presents some advantages of mobile cloud computing like high-level computing, secure storage, location independence, scalability, pay as you use, low total cost ownership, etc. Also, it presents a mechanism for providing confidentiality, access control, and integrity to mobile users.

Strengthening the authentication process depends on a strong authentication scheme and secure channel. Some classical techniques, like password based, one-time passwords, zero knowledge, and hashing, are used with modern techniques like biometrics.

In [10], they proposed a strong authentication method that is combined with providing identity management, mutual authentication, and session key establishment between the users and the cloud server. In [11], they attempted to deploy biometric authentication to increase their system's security. They applied the fingerprint as an input image to complete the authentication process.

In [12], the authors introduced the homomorphic encryption schema to preserve the privacy of the data in cloud computing by encrypting these data and allowing operations to be performed on these data without decrypting them. One of the homomorphic encryption schema is partial homomorphic, which allows only one operation to be performed, either addition or multiplication, on the encrypted data like RSA algorithms. These algorithms are not suitable for cloud computing because of their nature of allowing only one operation to be performed on the data. Therefore, fully homomorphic encryption is the solution to secure data in cloud computing. It allows for a number of operations to be performed many times. The authors mentioned DGHV and Gen10 cryptosystem as inefficient schema when used in cloud computing because the private key is needed to be sent to the server.

In [13], a secure and privacy-preserving digital rights management scheme (DRM) for cloud computing based on homomorphic encryption and proxy re-encryption is proposed to save the stored data and protect it in the cloud. The authors present DRM framework, which is efficient and capable of addressing the key management challenges. They provided a secured key distribution scheme based on an additive homomorphic probabilistic public key (AHPE) and proxy re-encryption (PRE). The user in their proposed scheme is not known to the server or service

provider. As a result, the computation complexity is reduced, and security level is raised.

3 Proposed System

The overall system design consists of user authentication and storage of data into the cloud. First is the authentication phase: user enters e-mail address and password along with fingerprint. They will be encrypted and stored in the cloud. In the log-in process, if the user's password or fingerprint after being encrypted matched with the encrypted identity on cloud, the user is granted authorization. Second, the system uses RSA encryption to create a secured channel between user and the cloud, so the user can establish a key to encrypt the data in the cloud without server's knowledge of the key.

3.1 Procedure

The overall system design, which has been implemented to increase security in mobile cloud computing, is shown in Fig. 1. The following are all steps needed to perform this experiment in details:

3.1.1 Phase 1: Initialization and Authentication

This phase is for generating user account through user e-mail and password and biometric authentication (fingerprint). AES cryptography is used, which has the ability to protect sensitive data from attackers, and it is able to have a high computational efficiency.

First, the user must *initialize an account* in the cloud computing. User initializes an account via secured channel; the process of initializing is as follows:

1. Establish a new account with e-mail, password, and fingerprint.
2. Key generation.
3. Use constant value to generate AES key.
4. Now the user can have the key.
5. Using AES 256, user password is encrypted and saved to the cloud, and user fingerprint is encrypted too.

In the case of log-in:

1. User encrypts password and fingerprint using user private key.
2. Submit the encrypted data to the cloud.

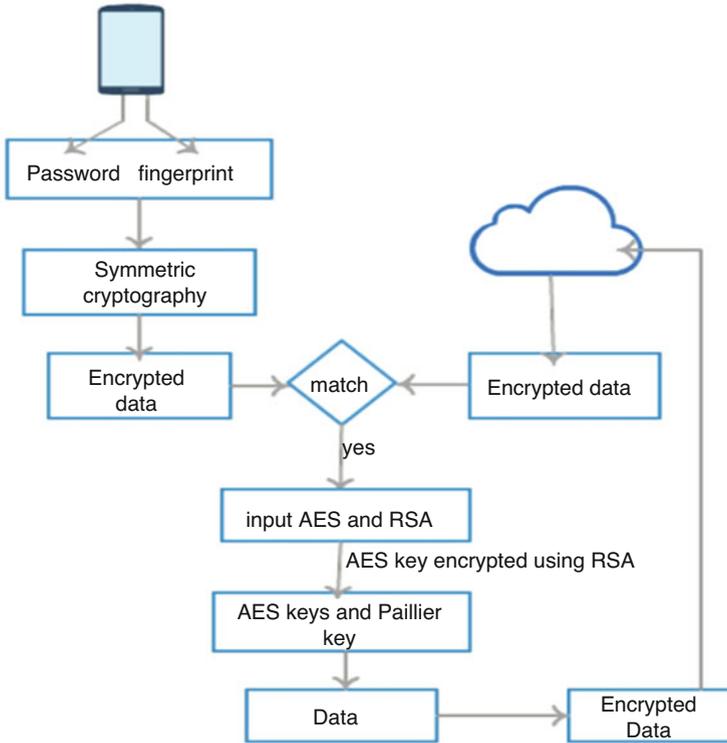


Fig. 1 Overall system design

3. Cloud server performs a matching search technique into stored encrypted data.
4. If they are matched, user will be authorized and granted access to the cloud; otherwise, access will be denied.
5. In the authorized access case, cloud will establish a secured channel by using RSA encryption algorithm to ensure security of key exchange and allowing the user to initialize his/her private key for the next phase.

3.1.2 Phase 2: Storing Data

When the authorization process is completed, the second step is achieved by generating the key to encrypt the data stored in the cloud as it is shown in Fig. 2. The system will establish a secured channel to exchange the key between user and cloud by using RSA cryptography. Then, the user will use it to generate his/her key by following this procedure. Therefore, server will not know the secret key of the user. For encrypting data in the cloud, symmetric homomorphic encryption will be using AES in control mode and Paillier homomorphic for mathematical operations. After

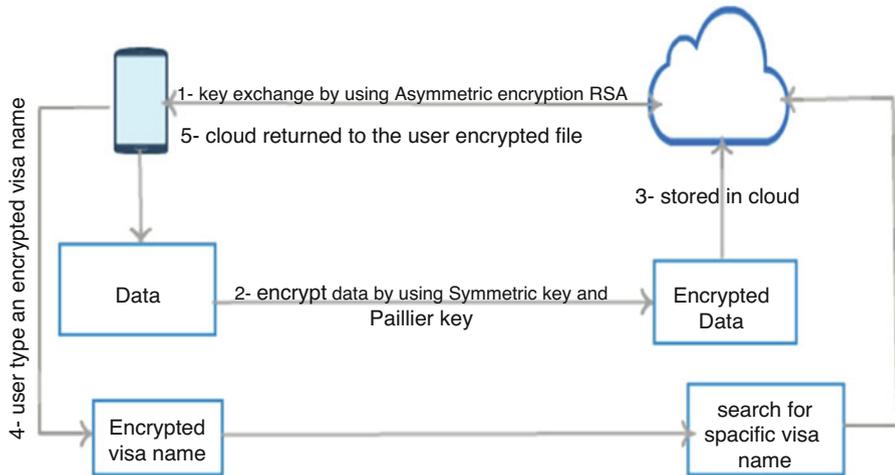


Fig. 2 Storing and process the data

that, he/she can add, view, or search the cloud for a specific file by typing its name to the server while it is in encrypted form. Whenever the client wants to decrypt the data, user can use his own private key to decrypt the data.

3.1.3 Secure Data

After the access is accepted

1. Cloud establishes a secured channel by using asymmetric encryption (RSA).
2. User encrypts data by using his/her key.
3. The encrypted data is stored at the cloud.
4. When user wants to perform an operation like search, user encrypts file name and then makes the search operation in the encrypted cloud server.
5. Cloud server returns the result to the client in encrypted form.
6. User can decrypt the returned file with his/her key.

3.1.4 Symmetric Homomorphic Encryption

The symmetric homomorphic encryption is proposed for many reasons: (1) to enhance the security, (2) minimize the implementation time and storage, and (3) improve the capacity of the homomorphic encryption algorithm.

Fig. 3 Registration page

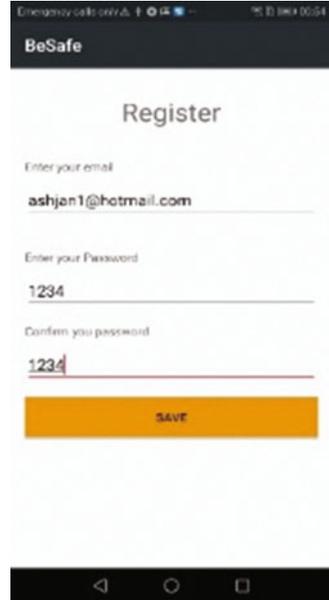
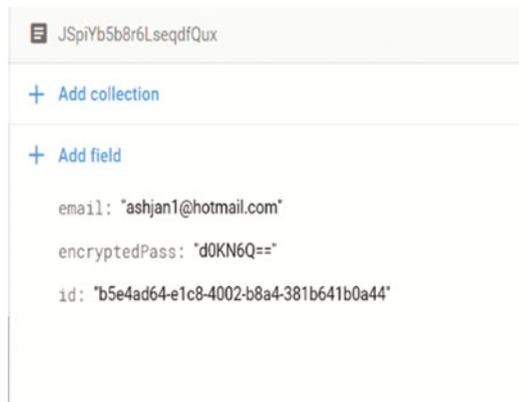


Fig. 4 User identity in the cloud



3.1.5 Software Model

Figure 3 shows the first phase of the system, which is the registration process. Figure 4 shows how the user identity is saved to the cloud. Figure 5 shows the application interface for adding data, deleting data, and searching for a specific data. Figure 6 shows how Visa credit card details are saved to the cloud.

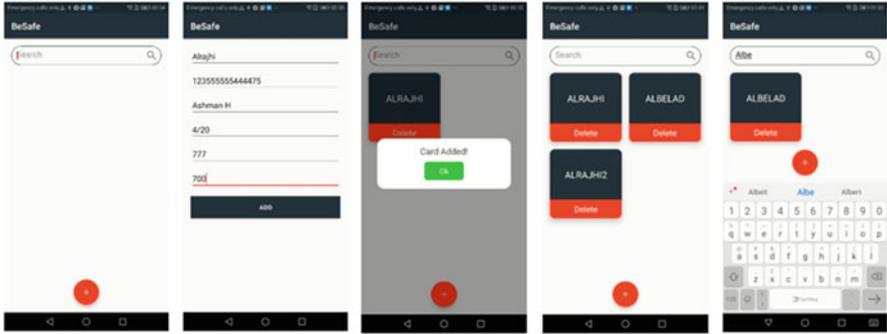


Fig. 5 Application interface

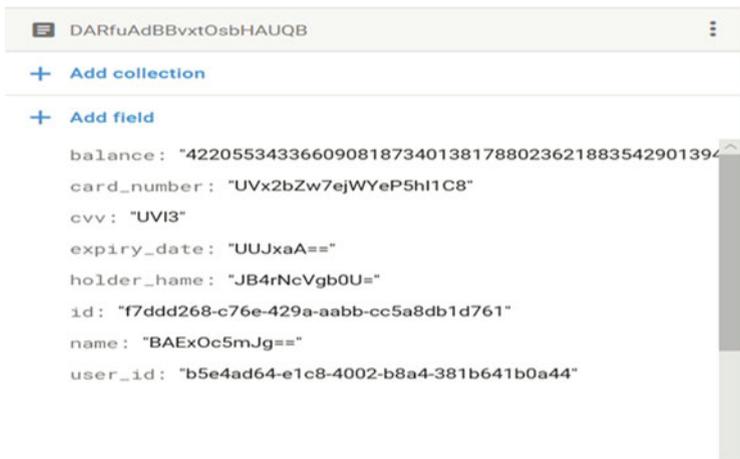


Fig. 6 Visa credit card details in encrypted form

4 Results and Discussion

An application has developed and implemented to enhance the security measures in mobile cloud computing. The overall system performance is the main concern as the user cares about the responsiveness of their devices. The application performance can be measured using CPU profiler, memory profiler, network profiler, and energy profiler.

Figure 7 shows the CPU utilization and thread state. As it is shown in Fig. 8, the virtual machine collects the call stacks of all the other threads in the process. The blue color is thread on run mode, while the yellow is in waiting mode. There are many threads running at the same time; the main one is the home activity.



Fig. 9 Call chart of view card activity

Table 1 Total time taken by threads

Encryption methods	Time in μ s
AES cryptography	1876
Paillier cryptography	2241
System performance	51,337

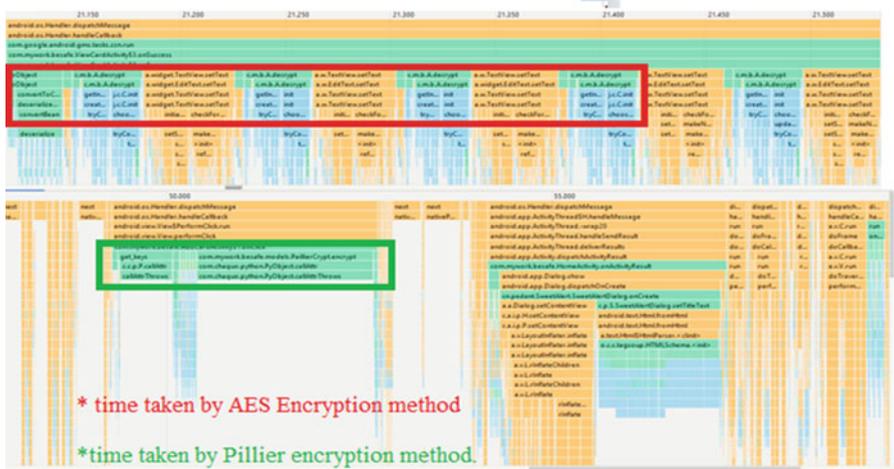


Fig. 10 Compression between Paillier and AES

performance is acceptable. As shown in Fig. 11, the marked peaks are the main jobs of the system, and they did not affect the performance as it is shown in the memory profile. Also, there is no object deallocation that effects battery life; therefore, the application did not consume the device battery too much.

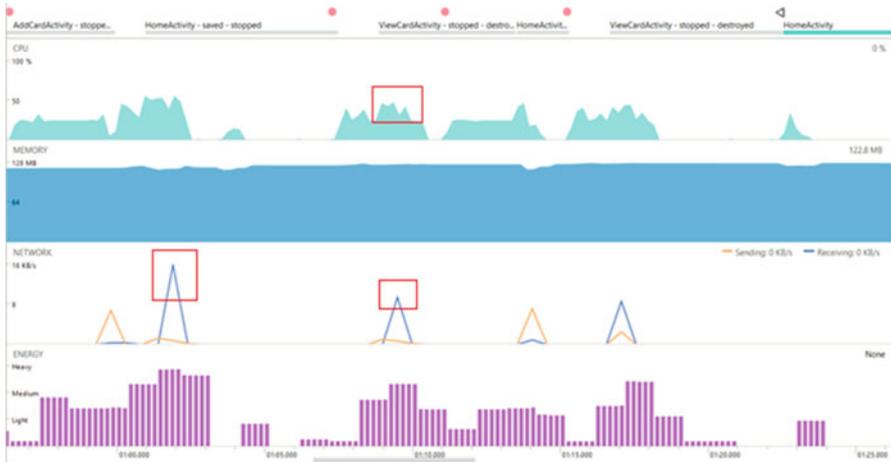


Fig. 11 Overall system performance

5 Conclusion and Future Works

In this work, we have proposed a system that secures user identity in mobile cloud computing and protects user data at the cloud server. This paper used hybrid technique by encrypting user’s password along with user’s biometric identification (fingerprint) and storing them to the cloud. AES counter mode has been applied along with Paillier homomorphic encryption that has the capability of partially homomorphic. The system consists of two phases, authentication phase and process phase. First, at the authentication phase, user password and fingerprint are encrypted to ensure the privacy of user identity and then send them to the cloud in order to apply the matching process. Second, after the user granted access to the system, user can encrypt his/her data by using user own private key and then upload these data to the cloud. Homomorphic encryption/decryption algorithm is used due to its advantages of protecting the privacy of the data and the ability to perform an operation on the encrypted data.

The challenge is remaining for securing the authentication and storage of data in mobile cloud computing. Therefore, securing the authentication to mobile cloud computing is as an open issue, and it is a great field for the researchers to dive in.

References

1. H. Chang, E. Choi, User authentication in cloud computing, in *International Conference on Ubiquitous Computing and Multimedia Applications*, (Springer, Berlin, Heidelberg, 2011), pp. 338–342

2. H.T. Dinh, C. Lee, D. Niyato, P. Wang, A survey of mobile cloud computing: Architecture, applications, and approaches. *Wirel. Commun. Mob. Comput.* **13**(18), 1587–1611 (2013)
3. S. Gawade, A. Bharti, A. Raj, S. Madane, Biometric Authentication using Software as a service in cloud computing. *Int. J. Eng. Comput. Sci.* **6**(3) (2017)
4. A. Huth, J. Cebula, The basics of cloud computing. *United States Computer* 1–4 (2011)
5. R.P. Kaur, A. Kaur, Perspectives of mobile cloud computing: architecture, applications and issues. *Int. J. Comput. Appl.* **101**(3) 0975–8887 (2014)
6. G. Reshmi, C.S. Rakshmy, A survey of authentication methods in mobile cloud computing, in *Internet Technology and Secured Transactions (ICITST), 2015 10th International Conference for*, (IEEE, 2015), pp. 58–63
7. M. Ahmadi, M. Chizari, M. Eslami, M.J. Golkar, M. Vali, Access control and user authentication concerns in cloud computing environments, in *Telematics and Future Generation Networks (TAFGEN), 2015 1st International Conference on*, (IEEE, 2015), pp. 39–43
8. N. Fernando, S.W. Loke, W. Rahayu, Mobile cloud computing: A survey. *Futur. Gener. Comput. Syst.* **29**(1), 84–106 (2013)
9. D. Tayade, Mobile cloud computing: Issues, security, advantages, trends. *Int. J. Comput. Sci. Inform. Technol.* **5**(5), 6635–6639 (2014)
10. A.J. Choudhury, P. Kumar, M. Sain, H. Lim, H. Jae-Lee, A strong user authentication framework for cloud computing, in *Services Computing Conference (APSCC), 2011 IEEE Asia-Pacific*, (IEEE, 2011), pp. 110–115fvv
11. P. Ruiu, G. Caragnano, G.L. Masala, E. Grosso, Accessing cloud services through biometrics authentication, in *Complex, Intelligent, and Software Intensive Systems (CISIS), 2016 10th International Conference on*, (IEEE, 2016, July), pp. 38–43
12. I. Jabbar, S. Najim, Using fully homomorphic encryption to secure cloud computing. *Internet Things Cloud Comput.* **4**(2), 13–18 (2016)
13. Q.L. Huang, Y.X. YANG, J.Y. FU, X.X. NIU, Secure and privacy-preserving DRM scheme using homomorphic encryption in cloud computing. *J. China Univ. Posts Telecommun.* **20**(6), 88–95 (2013)

Performance Analysis of Remote Desktop Session Host with Video Playback Scenarios



Baikjun Choi and Sooyong Park

1 Introduction

Many companies, public offices, and schools are configuring desktop as a service (DaaS) using Virtual Desktop Infrastructure (VDI) and Remote Desktop Session Host (RDSH). In addition, much effort has been made to provide DaaS in education or public web environment as well [1, 2]. One of the difficulties to configure such service environment is to configure a server resource in accordance with the number of users and use environment. Since the server resource is significantly different according to the constructionism each user uses, it is difficult to estimate the required resource. Thus, it is very hard to plan the acceptable number of users when developing DaaS.

Existing analyses of RDSH performance lack performance analysis data in relation to video playback. In the past, video playback was focused on entertainment rather than on work areas. However, it has been needed in work areas, as it now extends to information delivery areas, such as Internet lectures. Existing analysis on performance, excluding videos, is also not considered suitable for estimating the actual number of users, because multimedia content that includes videos is now emerging frequently in Internet browsing environments.

This study aims to estimate the acceptable number of users in a system by conducting a performance analysis based on scenarios that play videos.

B. Choi (✉) · S. Park
Sogang University, Seoul, Republic of Korea
e-mail: kjun@tilon.com

2 Related Study and Technologies

Studies in relation to server performance measurements targeting a large number of users were published as a form of white paper by Microsoft and included details tested over Windows Server 2003 and Windows Server 2008 R2 environments [3, 4]. Their studies modeled users' usage patterns for testing and defined them as "Knowledge Worker v2:"

- Preparing and storing documents using MS Word.
- Typing speed is 35 words per min.
- Document printout using MS Excel.
- Email check using Outlook.
- Slide addition and slide shows using MS PowerPoint.
- Web page browsing using Internet Explorer.

Table 1 presents the acceptable number of users according to scenarios and the capacity of the main memory in the Windows Server 2008 R2 environment [4]. Simulation tests were conducted based on the "Knowledge Worker v2" model defined for testing. The test results exhibited that the acceptable number of users was significantly different, according to the main memory size, in maintaining the sessions concurrently.

Remote desktop protocol (RDP) supports multimedia redirection (MMR) for videos and H.264/AVC encoding for graphic compression as well as hardware acceleration through a virtual graphic processing unit (vGPU).

3 Experiment Design

The experiment server was configured as presented in Table 2. The same performance comparison was conducted using "Knowledge Worker v2" in the experiment server to compare the results with existing ones. To test the multi session and RDP 10.0, the Windows Server 2012 and Windows 10 environments were used.

The experiment scripts were made, using MS Word, as document preparation scenarios while playing the video. For the video playback environment, the following three scenarios were prepared, considering the applied technologies.

- Preparing and storing documents using MS Word at a rate of 35 words per min

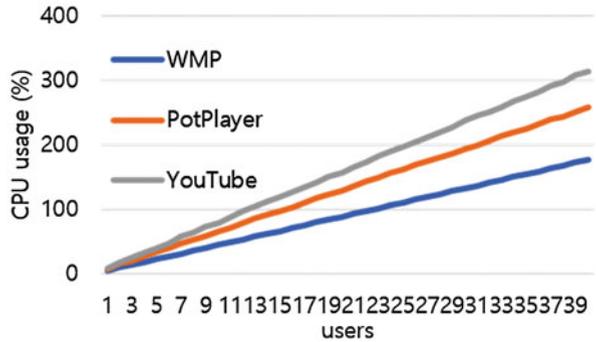
Table 1 Acceptable number of users according to scenarios and the capacity of main memory in the Windows Server 2008 R2 environment

	24GB memory	64GB memory	128GB memory
Knowledge Worker v2	80 users	150 users	310 users
With text-only presentation	Not tested	200 users	Not tested
Without PowerPoint	Not tested	230 users	450 users

Table 2 Configuration environment of experiment server

CPU	Intel Xeon E5-2640 2.50GHz * 2EA
Main memory	256GB
Disk	7200 rpm HDD in RAID 5
Network	1GB Ethernet
OS	Windows Server 2012/Windows 10

Fig. 1 CPU usage according to the increase in users



- Playing a 1280 * 720 video using three methods
 - Playback with Windows Media Player (WMP)
 - Playback with external player (PotPlayer)
 - Playback in the web player (YouTube)

The remote desktop load simulation tools [5] were used for session creation and for tests for experiments. For performance monitoring, the performance monitor in Windows was employed, in which central processing unit (CPU) usage, memory paging, disk activity time, and network utilization were monitored. Each of the video playback frame rates was also monitored, and the satisfaction of quality of service was defined as 20 frames per sec or higher. The redirection function of the USB, disk, and printer was all deactivated.

4 Experiment Result and Analysis

Since high utilizations of input/output (I/O) and CPU were exhibited during session creation for login, all users were logged in at least once to create a session before the experiment.

Figure 1 shows the CPU usage according to the increase in users in each scenario. Because no additional encoding process is required for the WMP where the Media Multimedia Redirection (MMR) was applied, the CPU usage was relatively low. However, the analysis result showed that YouTube had additional CPU usage due to the browser running.

Fig. 2 Network utilization according to the increase in users

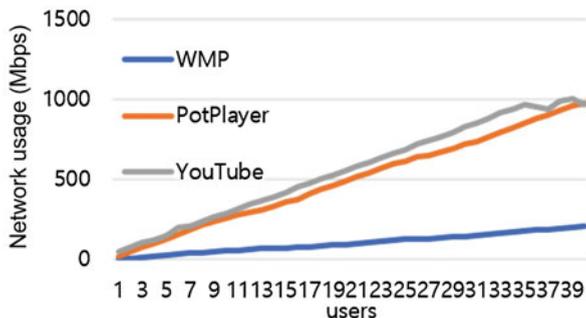


Table 3 Acceptable number of users in each scenario according to whether GPU is used

	CPU encoding	GPU encoding
Knowledge Worker v2	400 users	400 users
Word with WMP	100 users	100 users
Word with PotPlayer	30 users	40 users
Word with YouTube	30 users	35 users

The network utilizations of the video players are shown in Fig. 2. Since the MMR was applied to the WMP, the network utilization was relatively low. The network utilization of YouTube was higher than that of PotPlayer, which was analyzed as a result of the additional network utilization due to streaming. The network utilization changed as video playbacks of some users were suspended around 35 users in the YouTube scenario.

Around 500 users were estimated to be accommodated in “Knowledge Worker v2.” However, due to the disk I/O problem, only around 400 users could be connectible simultaneously. For “Word with WMP,” the network utilization and CPU usage were reduced as a result of the MMR application, resulting in the increase in the number of users at a level close to the network transmission limit. “Word with PotPlayer” and “Word with YouTube” could not maintain 30 users or more sessions due to CPU resource consumption as a result of real-time encoding. When applying encoding as well as GPU acceleration, the session could increase up to a level of 40 and 35 users, respectively, which was the allowable level of the network (Table 3).

5 Conclusion and Future Research

This study analyzed the performance of multiple RDSH sessions based on the video playback environments. The experiment results exhibited difficulties in session increase, due to the limitation of network resources. For future experiments, tests will be executed to help the configuration of the VDI environment through experimental designs considering the characteristics of a hypervisor along with 10 GB Ethernet.

References

1. S. Kibe, T. Koyama, M. Uehara, The evaluations of desktop as a service in an educational cloud, in *Network-Based Information Systems (NBIS), 2012 15th International Conference on*, (2012), pp. 621–626
2. A. Celesti, D. Mulfari, M. Fazio, M. Villari, A. Puliafito, Improving desktop as a service in OpenStack, in *Computers and Communication (ISCC), 2016 IEEE Symposium on*, (2016), pp. 281–288
3. B. Tritsch, *Microsoft Windows Server 2003 Terminal Services*: Microsoft Press (2004)
4. Microsoft, Remote desktop session host capacity planning in Windows Server 2008 R2, 2017. <https://blogs.technet.microsoft.com/enterprisemobility/2010/02/26/remote-desktop-session-host-capacity-planning-in-windows-server-2008-r2/>
5. Microsoft, Remote desktop load simulation tools, 05, 2017. <https://www.microsoft.com/en-us/download/details.aspx?id=2218>

Mining_RNA: WEB-Based System Using e-Science for Transcriptomic Data Mining



Carlos Renan Moreira, Christina Pacheco, Marcos Vinícius Pereira Diógenes, Pedro Victor Morais Batista, Pedro Fernandes Ribeiro Neto, Adriano Gomes da Silva, Stela Mirla da Silva Felipe, Vânia Marilande Ceccatto, Raquel Martins de Freitas, Thalia Katiane Sampaio Gurgel, Exlley Clemente dos Santos, Cynthia Moreira Maia, Thiago Alefy Almeida e Sousa, and Cicília Raquel Maia Leite

1 Introduction

Since the conclusion of the Human Genome Project in 2003, it was made possible that several studies could be carried out in the search for the cure of various diseases [1]. The use of bioinformatics contributed to the advance in the field and allowed the production of research in an accelerated pace. Currently, bioinformatics helps in the processing of huge amounts of results [2] and can also be used to perform data mining aiming at the extraction of relevant information that can help in the treatment or diagnosis of diseases [3]. The need to publish the huge amount of raw data from these studies led to the development of repositories where data could be made available in order to confirm the results or for further researches and may also allow for the inference of new information from the compilation of results from more than one study [4].

High-throughput gene expression (RNA) data, from microarray and RNA sequencing (RNA-Seq) studies, are mainly stored in three public databases: Gene

C. Renan Moreira (✉) · C. Pacheco · M. V. Pereira Diógenes · P. V. Morais Batista · P. F. Ribeiro Neto · A. G. da Silva · T. K. Sampaio Gurgel · E. C. dos Santos · C. Moreira Maia · T. A. Almeida e Sousa · C. R. Maia Leite
UERN, Mossoró, RN, Brasil
e-mail: marcosdiogenes@alu.uern.br

S. M. da Silva Felipe · V. Marilande Ceccatto · R. M. de Freitas
UECE, Fortaleza, CE, Brasil
e-mail: vania.ceccatto@uece.br

Expression Omnibus¹ (GEO), ArrayExpress² (AE), and Genomic Expression Archive³ (GEA). Although there is a centralization of this information in the abovementioned repositories, there is still some difficulty in conducting studies in the research metadata. This is due to the lack of standardization in the vocabulary employed by the authors of the researches to structure the data that will be stored in the databases [5]. Many researchers use algorithms written in the R programming language for the analysis of this data; however this can be a problematic factor since it is not simple for the life science research community to use or write codes in some programming language. Bioconductor⁴ is a repository that provides several tools for this purpose [6]; however the lack of programming skills may drive some researchers away.

To mitigate this situation, the present work proposes, through the application of an e-Science approach, the development of a WEB system is capable of reading a massive amount of data, pre-processing, mining, and displaying it in a user-friendly interface, intending to aid researchers to delve further into the existing research. The architecture and functionalities proposed for this system and the preliminary results of the research will also be demonstrated.

2 e-Science and Biological Data Mining

With the amount of raw data constantly been generated, it would be a complicated task to process this data manually. Being able to combine the raw microarray analysis data from two or more studies would result in a huge effort on the part of the researchers. In order to facilitate such analyses, the scientists can make use of bioinformatics, using e-Science to process a massive amount of data, and can also make use of data mining techniques to be able to extract new information from the pre-existing surveys.

The use of e-Science is justified by offering the researcher a platform capable of storing, interpreting, analyzing, and making available in network this data to enable other working groups to make use of this. With the use of the technique, it will be possible to deal with large volumes of data in a properly scalable system. This will enable the accomplishment of research in a quick and efficient manner, also enabling joint and multidisciplinary studies in the context of bioinformatics.

One of the approaches that can be used in bioinformatics is the mining of biological data [7]. Data mining uses algorithms in the search for valid and understandable patterns within the available data and includes the following steps: associating, grouping, and discovery of classification rules [8]. It is a fundamental

¹<https://www.ncbi.nlm.nih.gov/geo>.

²<https://www.ebi.ac.uk/arrayexpress/>.

³<https://www.ddbj.nig.ac.jp/gea/index-e.html>.

⁴<https://www.bioconductor.org/>.

technique in bioinformatics because it allows researchers not to be mere observers of the available platforms (e.g., ENSEMBL and UCSC Genome Browser) and explore the information contained within the data [3]. Data mining can help transform data that is not easily understandable through a purely visual interpretation and can also facilitate the inference of new and useful information for the discovery of new results from new or pre-existing research [9]. The new results from the existing research could be achieved through the researchers' new approach to the data previously analyzed in another study. Often these datasets are deposited in public biological databases and lay there almost forgotten, and new analyses could provide advances in their respective research areas [10].

Currently one of the possible ways to conduct biological data mining is through bioconductor [6], a project based on the premise of free software that aims to promote statistical analysis and understanding of high-throughput data from new and pre-existing biological studies. The project is based on packages written largely using the R programming language; however it may contain contributions from other languages. Bioconductor is one of the most relevant repositories of tools for the study of biological data; the packages made available are destined toward various types of analyses, among them, data mining.

3 Description and Context of the Proposed System

In recent years, data from scientific research in the biological area have been made available in public databases dedicated to the storage and display of this information [11]. At a very similar rate, computational tools capable of interpreting biological data have been emerging in recent decades in order to evaluate the data and produce new results [5].

One of the most popular transcriptomic databases in the scientific community is Gene Expression Omnibus. Its platform has almost 4,500 studies stored in the form of datasets, from researches that analyzed almost 3.5 million samples to generate their results [5]. Currently, the way the platform makes this data available is not easily manageable for users without appropriate technical knowledge for reading information in plain text files, having their data divided online and separated by tabs. Therefore, it is necessary to use tools that can help in this setback so that the researcher can dedicate his effort to the data and not to understanding the storage model.

This research intends to implement a system capable of providing a service that recovers data from the GEO platform; this data will be pre-processed so that the information from the platform is validated in order to pass correctly to the later steps that may include, according to the user's preference, mining of this data or manual processing of information from filters made available in the system being developed.

During the development of the proposed system, as well as in the subsequent stages, after implementation, the objective is to achieve the following results:

1. Implement an Application Programming Interface (API) that allows interaction between the GEO biological database and a local database.
2. Develop an interface for pre-processing obtained data.
3. Implement data mining and machine learning techniques to act upon the pre-processed data.
4. Develop an analytical interface and the interaction between the researcher and the studies' data.
5. Allow that researchers without computer programming knowledge can utilize data from the biological databases.
6. Provide the possibility to extract new information from raw data made publicly available in the GEO biological database.
7. Develop a system that can be used in the daily life of researchers in the area of biological and health sciences.

To achieve the abovementioned results, we intend to develop a WEB system that will be available full-time in a domain on the Internet, thus enabling researchers from all over the world to use it. Initially, it is intended that the system be available in two languages, Portuguese and English. The use of the system should follow the step-by-step style where the researcher should advance between the screens until he reaches the listing of the filtered data during the previous steps.

With the data adequately filtered and the results exposed to the researcher, the system will provide a series of filters aimed at the selection of useful data, to try to identify relevant information aiming for new scientific discoveries. Among the metrics that can be used in the analysis of the results is fold change, this metric evaluates how much higher or lower a given gene was expressed by comparing samples from two individuals or two groups of patients within a given research.

Initially the tests and filters will be applied in studies related to diabetes from the biological database GEO. In this initial stage, compatibility with studies that used the DNA microarray technique will be offered.

4 System Architecture

As can be observed in Fig. 1, the architecture planned for the system will have three distinct APIs, a database, and a WEB interface in which users will be able to query the data made available by the system. This division into distinct APIs and interfaces was motivated by the possibility of allocating the algorithms on different servers if necessary. Some algorithms that are part of the APIs can be computationally complex, and this can lead to a very high processing load. In order to ease the process, aiming for everything to be better tiered, the decentralized architecture presented itself as an efficient solution.

API 01 will be responsible for the tasks related to mining the obtained data, as well as the machine learning tasks. It is intended that this API will support the implementation of several mining algorithms that will be useful for analyzing the

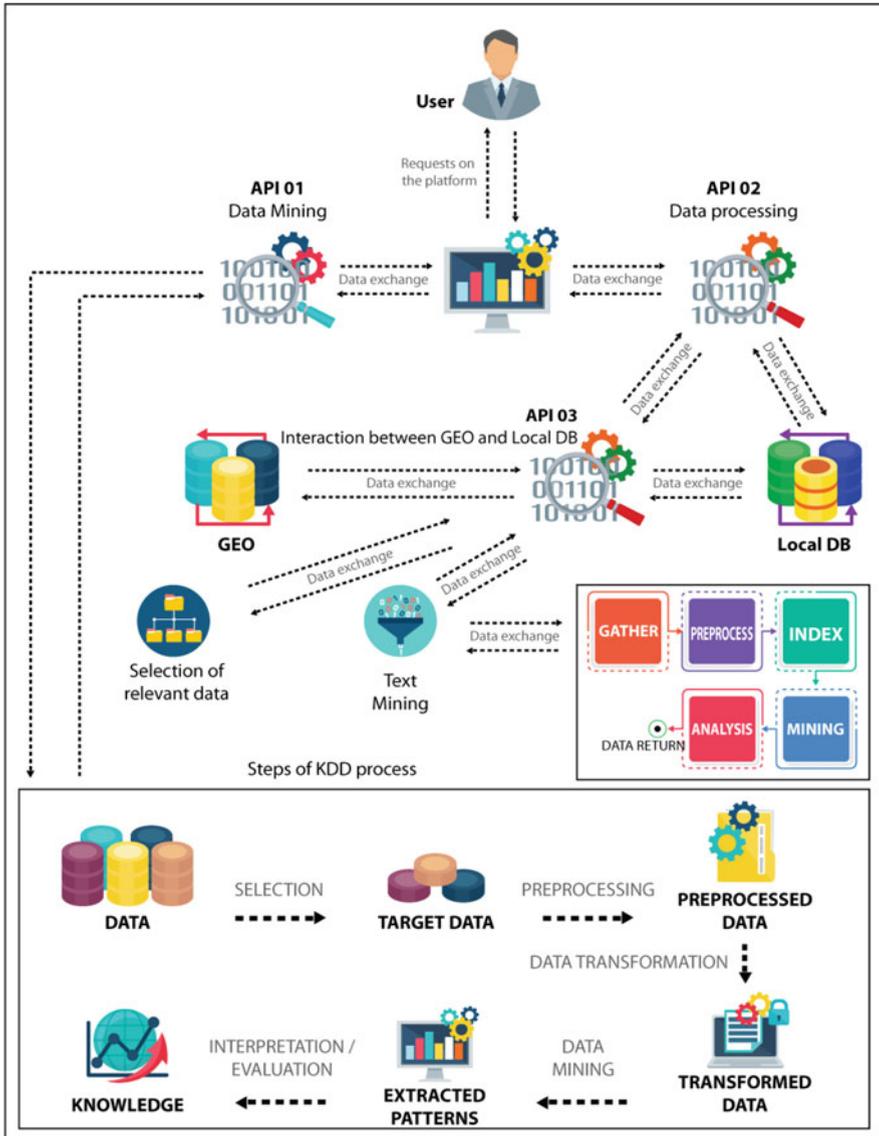


Fig. 1 System architecture overview. Source: The authors

data of the study explored by the scientist. The algorithms should be implemented gradually, thus supporting improvements even after the completion of the initial phase of the project.

API 02 will be responsible for pre-processing the data that will be uploaded to the WEB system; the API in question can also load data directly from the local data pool, as well as request API 03 to search for new data in the GEO biological database. The development of this API must strictly respect principles related to reuse so that new filters can be inserted into future implementations of the project.

API 03 will capture the data from the dataset requested by the user and store the relevant information in the local database. Although it is a simple task, sometimes considerable processing power is required because the data from some searches is too complex, with a lot of information. The API will interpret all the data and store it so that later processing steps are faster.

The WEB system will include the interface that will receive all user requests. The procedures to be performed should be moderately facilitated, but important filters will not be left aside because they are complex; however their use should be intuitive and whenever possible, with mechanisms to help the user. It is intended to be an adaptive system, but at the moment, a specific interface for devices with small resolution is not part of the scope. This choice is justified by the density of the data to be displayed, thus requiring a higher resolution for an intelligible visualization.

The system's users need to be registered so that in this way, there can be adequate customization, as well as enable relevant analyses for decision-making related to resource allocation and future implementations. This authentication is also intended to prevent user abuse in order to preserve the availability and integrity of the developed system.

Finally, the local database will be relational, and it will store all the data that is relevant to the queries. It is intended that some processed data is stored in a way that allows the user to retrieve the results processed by him, he should also store presets of filters defined by the user so that he can reapply the same filters in future situations.

5 Preliminary Results

During the studies for the development of this project, it was necessary to carry out the implementation of some functionalities that will be part of the system. In order to perform the initial analyses of the information available on the GEO platform, a preliminary study was carried out in order to map the format and patterns of the data contained in the files made available by the GEO database. From this starting point, it was possible to create an intelligent search algorithm capable of reading and interpreting datasets made available by GEO and store the relevant data collected. A relational database was designed for storing information.

In order to visualize the results, a WEB interface was created, in which researchers can define which data within the datasets they wish to use in their

searches. As an output, the system will show relevant data in a user-friendly interface. Within each dataset, it is possible to visualize information about how the original gene expression study was conducted, how the study divided the subjects into subgroups, as well as the gene expression readings of the subjects studied.

In Fig. 2, it is possible to visualize some data obtained from a study on diabetes. The first table, denominated dataset, contains some basic information about the research such as the search code on the GEO platform, the title, a description, and the organism in which that research was conducted. It is also possible that, through the “See Subsets” link, the groups of patients who were part of the research are viewed. In the second table it is possible to identify some data obtained in the original microarray study. The columns initiated with “GSM” show the gene expression of individuals participating in the research for different genes, these are identified in the “Identifier” column.

Since the presented data are divided into research groups, it is possible to perform the comparison between the groups so that the fold change calculations are performed, allowing to evaluate which gene set has a significant change when comparing the control group and a group of patients with a certain disease. Also in this study, it was possible to obtain a series of other numerical or categorical data related to the research; these data can be mined in search of relevant information. It was also perceived that the texts described by the authors of the researches do

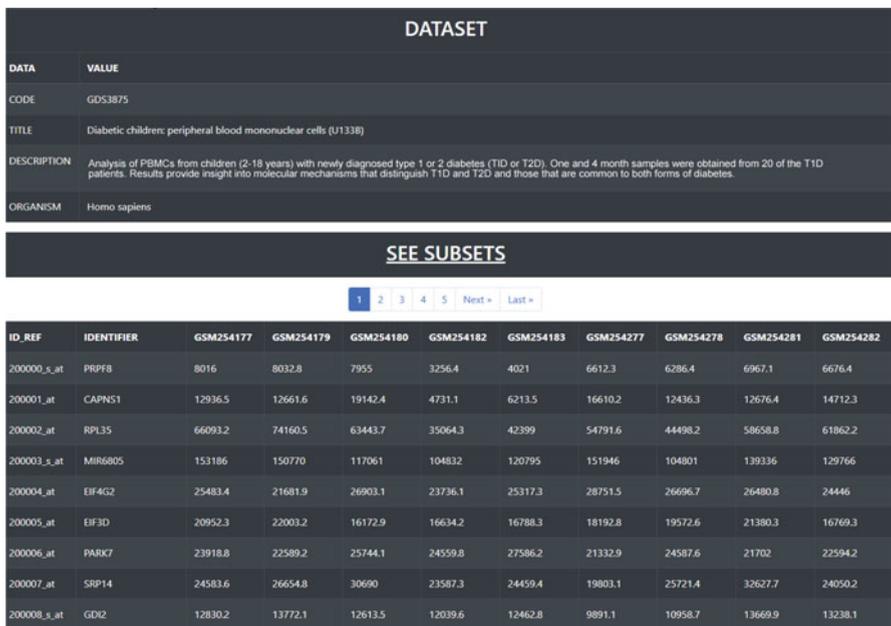


Fig. 2 Partial visualization interface for obtained data. Source: The authors

not have a well-defined structure and, with this it was perceived that an automated analysis of this information could only be made from the use of text mining.

The data obtained from the metadata of the captured GEO datasets demonstrated that the objectives foreseen for this study are achievable and also shows that other functionalities can be implemented in future versions of the project, thus ensuring that a more complete system can be made available to the target audience.

6 Conclusion

This paper describes a system under development capable of assisting biological and life sciences researchers in conducting new studies from public data from various scientific researches. Initiatives like this enable computing to support new discoveries without the need to conduct new laboratory tests. The results already obtained prove that with due treatment the data can be easily interpreted and thus easing the difficulty of comprehension of these data on the part of researchers.

In future developments, we intend to adapt the system in order to include research data that use the RNA-Seq technique, so that more studies can be analyzed. It is also part of future implementations to enable the system to obtain data from other biological databases.

Acknowledgments The authors would like to thank *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)* and *Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq)* for funding researchers' scholarships.

References

1. K.-C. Wong, Big data challenges in genome informatics. *Biophys. Rev.* **11**(1), 51–54 (2019). <https://doi.org/10.1007/s12551-018-0493-5>
2. M. Gasparovica-Asite, L. Aleksejeva, Classification methodology for bioinformatics data analysis. *Autom. Control. Comput. Sci.* **53**(1), 28–38 (2019)
3. X.M. Fernández-Suárez, E. Birney, Advanced genomic data mining. *PLoS Comput. Biol.* **4**(9), e1000121 (2008), 18818719[pmid]. <https://www.ncbi.nlm.nih.gov/pubmed/18818719>
4. A. Brazma, Minimum information about a microarray experiment (miame)—successes, failures, challenges. *TheScientificWorldJournal* **9**, 420–3 (2009)
5. Z. Wang, A. Lachmann, A. Ma'ayan, Mining data and metadata from the gene expression omnibus. *Biophys. Rev.* **11**(1), 103–110 (2019). <https://doi.org/10.1007/s12551-018-0490-8>
6. H. Nie, P. Neerincx, J. van der Poel, F. Ferrari, S. Biccato, J. Leunissen, M. Groenen, Microarray data mining using bioconductor packages. *BMC Proc.* **3**(Suppl. 4), S9 (2009)
7. V. Gangwar, U. Ghose, Y. Singh, Data mining of biological data in bioinformatics using transcription, translation algorithm and pattern matching of protein sequences. *Int. J. Adv. Res. Comput. Sci.* **3**(3) (2012)
8. F.S. Espindola, L.K. Calábria, A. Azenha Alves de Rezende, B. Barbosa Pereira, F. Assumpção Santana, I. Marques Rodrigues Amaral, J. Lobato, J. Luzia França, J. Luiz Mario, L. Bruno Figueiredo, L. Pereira dos Santos, N. Moura de Gouveia, R. Nasci-

- mento, R. Roland Teixeira, T. Alves dos Reis, T. Gonçalves de Araújo, Recursos de bioinformática aplicados às ciências ômicas como genômica, transcriptômica, proteômica, interatômica e metabolômica. *Biosci. J.* **26**(3) (2010). <http://www.seer.ufu.br/index.php/biosciencejournal/article/view/7146>
9. S.B. Garg, A.K. Mahajan, T. Kamal, An approach for diabetes detection using data mining classification techniques. *J. Eng. Sci.* **26** (2017)
 10. K. Lan, D.-T. Wang, S. Fong, L.-S. Liu, K.K. Wong, N. Dey, A survey of data mining and deep learning in bioinformatics. *J. Med. Syst.* **42**(8), 139 (2018)
 11. H. Bono, All of gene expression (AOE): an integrated index for public gene expression databases. *PLoS One* **15**(1), e0227076 (2020)

Index

A

- ABC classification model, 842–844, 850, 851, 856
- Accuracy, 382–383
 - active learning algorithm, 609
 - anatomical variability, 343
 - continuous enlargement, 56
 - dark data extraction, 100
 - dielectric constants, 52
 - machine learning, 56–61
 - python block, 87
 - QR algorithm, 698
 - rotation, 715–717
 - and speed, 714
- Active learning
 - deep learning methods, 613–615
 - machine learning, 609
 - pool-based, 610–611
 - practical and real-world problems, 609
 - query methods
 - estimated error reduction, 613
 - QBC, 612–613
 - uncertainty sampling, 612
 - query synthesis, 610
 - stream-based, 610
 - survey, 610
- Actor-based containers (Abaco) platform
 - autoscaling, 1061
 - comprehensive test, 1062–1063
 - containers-as-a-service, 1075
 - distributed computing platforms, 1075
 - experiment design
 - overview, 1062
 - research questions, 1061–1062
 - scaling test, 1062
 - experiment setup
 - resource configuration, 1065
 - resource deployment, 1066
 - validation, 1067–1068
 - validation setup, 1067
 - functions-as-a-service, 1074–1075
 - node sizes, 1071–1073
 - obtaining theoretical bounds, 1064–1065
 - performance metrics
 - FLOPS, 1063
 - Hashrate, 1064
 - RabbitMQ, 1060
 - rate of worker creation
 - autoscaler case, 1069, 1070
 - autoscaler rate, 1071
 - comparison, 1069, 1070
 - manually scaled case, 1069
 - resource requirements, 1060
 - scaling limits, 1072, 1074
 - stateless, 1060
 - UNIX domain socket, 1060
 - users, 1059
 - worker configurations, 1068
- Actor Model of Concurrent Computation, 1059
- Adaptive tempering Monte Carlo (ATMC)
 - method, 248, 252–254
- Advanced persistent threat (APT), 36, 41
- Agent-based modeling, 579
- Agile programming, 307, 314–315
- AlexNet, 87
- Algorithm decoupling, 408–409

- Algorithm flattening (AF)
 - branch elimination, 650–654
 - generalized preliminary, 652
 - optimized and reduced AF, 652–653
 - utilization, 648
 - Amazon’s Elastic Container Service (ECS), 1075
 - Analytic hierarchy process (AHP), 842
 - consistency calculation, 847–849
 - qualitative and multivariate values, 849–851
 - weights calculation, 846–847
 - Analytics
 - AJ matrix, 380
 - ASUM, 68
 - end-to-end, 636
 - neutrinos, 176
 - reports/survey data, 92
 - sequential data flows, 406
 - technologies, 98
 - trusted scalability, 634
 - Analytics Solutions Unified Method (ASUM), 65, 68, 69
 - Analyzer module, 388, 389, 391–393, 397, 403
 - Animations, 933–934
 - Anomaly detection, 782
 - Apache Spark, 1075
 - Apache Storm, 1075
 - Application model, 388, 391, 393, 395–397
 - Application programming interface (API), 180, 419, 479, 485, 596, 600, 1059, 1061, 1078
 - APT, *see* Advanced persistent threat (APT)
 - Asian Integrated Model (AIM), 861
 - ASUM, *see* Analytics Solutions Unified Method (ASUM)
 - Attacks
 - Android app store, 31
 - example, 33–34
 - lock screen, 25
 - ransomware, 26
 - Trojan, 19
 - user’s computer, 18
 - watering hole, 36
 - website, 22
 - See also* Cyberattacks
 - Augmented implicitly restarted Lanczos bidiagonalization (AIRLB) algorithm
 - and GKL, 701
 - improvements, 701–703
 - Krylov subspace method, 701
 - lithography simulation modeling, 698
 - numerical experiments
 - computation results, 708–711
 - experimental environment, 706–708
 - preconditioning technique, 701
 - regularization technique, 697–698
 - Augmented reality (AR), 747–755
 - Authentication
 - and authorization, 109
 - biometric, 1180
 - bypass, 311
 - and identification, 364
 - initialization and, 1180–1181
 - message, 309
 - OpenStack, 1066
 - system’s security, 1179
 - Autodesk Maya, 829
 - Automation
 - base system, 1018
 - basic resources, 1015
 - control algorithm, 1019
 - effects, 1017
 - electric lights, 1015
 - energy generation, 1021
 - fan control system, 1019
 - internal temperature, 1020
 - lighting schedules, 1018
 - materials, 1016
 - methods, 1016
 - number of devices, 1019, 1020
 - off-grid farming systems, 1016
 - physical systems, 1022
 - PLC, 1018
 - power, 1017
 - power stability, 1018
 - racks, 1017
 - Raspberry Pi network, 1018, 1021
 - renewable energy, 1015
 - simulation, 1021
 - vertical farming systems, 1015
 - water pumps, 1015
 - Average waiting time, 547, 553, 554
 - AWS Kinesis, 1075
- B**
- Band gap prediction, 760–767
 - Barcode scanners, 325
 - Bayesian Graphical Lasso (BGL), 781, 783
 - Bayesian information criterion (BIC), 851
 - Bayesian sparse covariance structure analysis
 - Bayesian Graphical Lasso (BGL), 783
 - crime spots data, 789–790
 - Graphical Lasso, 782
 - Poisson process and crime data, 783
 - proposed method, 784–785

- sampling scheme, 785
 - synthetic data analyses, 786–789
- Berendsen temperature, 1024
- Bicubes
 - cube-based topologies, 527–530
 - definitions, 525–527
 - even-dimensional bicube, 537–538
 - interconnection networks, 527–530
 - odd-dimensional bicube, 531–537
 - topologies, 525
- Big data, 862
- Binary images
 - inverse problems, 325
 - line
 - in image, 326
 - thickness, 329
 - orientation determination, 328–333
 - previous work, 326–328
 - Radon transform, 325
 - 2D image intensity function, 326
- Bioinformatics problems, 737–738
- Biological data mining, 1197
- Bitcoin, 1060, 1064
- Bladerunner, 294, 296, 298, 305
- Blockchain technology
 - data collection, 125
 - identifiers, 120
 - implementation
 - code snippets, 129–130
 - future studies, 130
 - literature review
 - in reputation system, 121–122
 - reviews and reputation systems, 120–121
 - members' opinion, 119
 - peer-to-peer system, 119
 - system analysis results
 - quantitative data analysis, 125
 - system design, 126–128
 - system requirements, 125–126
 - trusted reviews model, 123–125
- BlueBox2.0
 - development platform
 - Cortex-A72 layer, 78
 - hardware architecture, 78, 79
 - implementation results, 86–87
 - LS-2084A, 80
 - RTMaps, 80–81
 - vision processor (S32V234), 78, 80
- Bond charge correction (BCC) method, 1024
- Branch elimination, 648, 650–654
- Branch splitting, 648, 649
- C**
 - C#, 831
 - Cadence software, 881
 - CAD solid model, 829
 - Caloric curve, 248, 253, 1024
 - Capacitated minimum spanning tree (CMST) problem
 - computational study, 356–360
 - cross decomposition, 349–353
 - edge exchange heuristic, 353–356
 - multicommodity flow formulation, 351–352
 - network design, 347
 - single-commodity flow formulation, 348
 - Card deck model, 962
 - CAT, *see* Computed axial tomography (CAT)
 - Center for internet security (CIS), 322–323
 - Central processing units (CPU), 108, 405, 493, 1191
 - Apple MacOS Mojave 10.14.6., 500
 - capabilities, 406
 - comparison
 - programming examples, 502–504
 - qualitative observations, 504–505
 - quantitative measurements, 499–502
 - computing devices, 493
 - data flow, 407
 - FPGA-GPU-CPU heterogeneous system, 477–491
 - vs.* GPU, 427, 466
 - GPU-accelerated benchmarks, 426
 - hardware architectures
 - Core i7, 496
 - Intel Xeon W-2191B, 497
 - TPA, 495–496
 - Intel Skylake Xeon W CPU, 493, 494
 - machine code, 5
 - and memory, 108
 - multi-core, 740
 - OpenCL-enabled Altera Stratix V FPGA, 473
 - parallel process, 446
 - peak usage periods, 314
 - performance and programmability, 494
 - programming methodologies
 - POSIX threads (pthreads), 498
 - TCF, 497–498
 - resources and RAM, 11
 - serial run, 413
 - Skylake, 505
 - CGT, *see* Computerized geophysical tomography (CGT)

- Channel
 - communication, 1161
 - computational fluid dynamics, 185
 - digital, 955
 - I/O, 470–471, 474
 - RSA encryption algorithm, 1181
 - step, 187, 193
- Character recognition, 685, 686, 689, 690, 694
 - See also* Japanese character recognition
- Chronic obstructive pulmonary disease (COPD), 834
- Chrono, 875
- CIS, *see* Center for internet security (CIS)
- Cloud computing
 - abnormal detection rate, 1133
 - algorithm-based probability theory using SAS, 1131
 - cloud and edge computing (*see* Cloud-edge centric IoT architecture)
 - DTBAC model, 1121
 - FMUBCT model
 - abnormal detection rate, 1133
 - authentication module, 1122
 - authorization module, 1122
 - behavior monitoring module, 1122
 - comparison module, 1123
 - flowchart, 1123
 - fuzzy logic module, 1123, 1127, 1130
 - rapid decrease and slow-rise strategies, 1133
 - trust computation module, 1124–1129
 - trust dynamic management module, 1124
 - trust types, 1132, 1133
 - user profiling module, 1123
 - framework, 623
 - fuzzy AHP model, 1121
 - models performance, 1133, 1134
 - multiple servers, 619
 - protein–protein docking, 737–744
 - RPTM, 1121
 - RSUs, 541
 - SaaS application, 1121
 - servers, 307
 - and devices, 619
 - services, 619
 - trust value evaluation principles
 - expiration records, 1129–1130
 - malicious behavior, 1131
 - rapid decrease strategy, 1131
 - recent behavior principle, 1130
 - slow-rise strategy, 1130
 - UBADAC model, 1122
 - UBTMFL model, 1122
- Cloud-edge centric IoT architecture, 620
 - cloud layer, 624, 625
 - edge layer, 624, 625
 - experiments, 628–629
 - IoT layer, 624–626
 - task assignment, 627–628
 - task model, 626
 - VM model, 626
- Cloud providers
 - cybersecurity, 308
 - data center locality, 316
 - encryption
 - at rest, 317–318
 - in transit, 318–319
 - non-linear pricing scheme, 1167
 - problem statement and discussion
 - agile programming, 314–315
 - security support, 315–316
 - security training for employees, 315
 - recommendations
 - CIS, 322–323
 - cloud security comparison data, 321–322
 - multiple, 320
 - security of the cloud, 321
 - related work, 308–309
 - security threats, 307, 309–314
 - top three cloud platforms, 319
 - uptime/availability of platform, 316–317
- Cloud security
 - comparison data, 321–322
 - controls, 309
 - security installation and enforcement, 315
- Cloud Security Alliance (CSA), 309
- Cloud service providers (CSPs), 1033
- Clustering, 66, 177, 634, 795–796
- CMST problem, *see* Capacitated minimum spanning tree (CMST) problem
- CNN, *see* Convolutional neural networks (CNNs)
- Code snippets, 129–130
- Collaborative applications, 659
- Command line interface (CLI), 1065
- Common Attack Pattern Enumeration and Classification (CAPEC), 580, 581
 - online database, 581
 - PNPSCs, 581–582
- Compensators, 237
 - MOWOA and AHP method, 240
 - multi-DG, 235, 244
 - power electronic devices, 235
 - serial, 819
- Composability, 580
- Computed axial tomography (CAT), 325

- Computerized geophysical tomography (CGT), 325
- Computer science (CS)
 - computing tests, 134
 - interception, 65
 - natural sciences and engineering, 486
- Conceptual model, 843
- Condensed polymer systems
 - atomic charge refinement, 1024
 - density functional theory, 1024
 - force field, 1024
 - glycolic acid (G), 1024
 - gyration, 1026, 1027
 - isobaric analysis, 1024
 - isochoric analysis, 1026
 - lactic acid (L), 1024
 - Legendre transform, 1027
 - linear regions, 1025
 - molecular dynamics (MD), 1024
 - monomer sequence, 1024
 - nanoparticles, 1028
 - nanostructures, 1028
 - NPT annealing, 1024, 1025
 - NVE production, 1026
 - PLGA, 1023
 - PLGA(50:50) properties, 1027, 1028
 - polymeric nanoparticles, 1023
 - polymer system behavior, 1026
 - specific heat and thermal expansion, 1025
 - structural analysis, 1026
 - structural optimization, 1024
 - thermal compressibility coefficient, 1027
 - volume selection, 1026
- Containerization
 - containers and virtual machines, 11–12
 - Docker vs. Singularity, 13
 - Singularity and GPUs, 13–14
 - virtualization, 313
- Control divergence, 648
- Conventional VR devices, 749
- Convolutional neural networks (CNNs), 613, 769
- Core i7, 495, 496, 500–503, 505
- Counter attack, 46
- Coupling coefficient, 773
- Coventor MEMS+ software, 881
- Coventor software, 886
- COVID-19
 - blacklisted directory, 34
 - crisis, 953
 - digitization, 944, 956–957
 - economic crisis, 943
 - global pandemic, 25
 - heatmap visuals, 33
 - impact of, 944
 - sustainability and social responsibility, 949, 950
- CovidLock
 - application, 30
 - lock screen attack, 25
 - ransom lock screen, 31
 - ransomware, 32
 - run-times, 28
- CPEs
 - data allocation, 139–141
 - hotspot functions, 147
 - large-scale
 - parallel computing, 134
 - parallelization, 139
 - memory hierarchy, 135
 - updated lattice data, 142
- CPU, *see* Central processing units (CPU)
- Crawling technique, 684, 688–691, 694
- CREATE-GV Mercury software tool, 888, 890
- Create, Read, Update and Delete (CRUD) operations, 1078, 1082
- Crime occurrence, 782
- Crime spots data
 - spatial crime data, 789
 - visualization of partial correlations, 790
 - Ω partial correlation, 789–790
- Critical thinking, 928
- Cube-based topologies, 525, 527–530
- cuBLAS library, 419
- Cultural competence training, 929
- Cyberattacks, 32, 35, 586, 589, 590, 625
 - CAPEC resource, 580, 581
 - composability, 580
 - malicious, 35
 - parent–child PNPSC models, 589
 - PNPSC, 581 (*see also* Petri Nets with Players, Strategies and Cost (PNPSC))
- Cybersecurity
 - active research, 579
 - CAPEC, 581
 - Cloud IaaS, 315
 - computer networks, 581
 - cyber-physical vehicle systems, 153
 - detection of, 43
 - game theory, 579
 - hardware and software operations, 100
 - IaaS environment, 311
 - issues/attacks, 43
 - modeling, 579
 - monitoring, 97
 - parallel processing, 438
 - password hashing, 438–440

- Cybersecurity (*cont.*)
 risk, 579
 SAE J3061, 153
 SYN flood attacks, 43
- D**
- DaaS in mobile environment
 effective transmission methods
 existing compression methods, 1138, 1139
 network structure, 1139
 screen delay, 1142
 server and client structure, 1140
 system structure, 1140–1142
 selective compressed codec
 codec delay, 1145
 compression method, 1144
 DCT-based compression method, 1143
 H.264 decoding hardware, 1145
 HuffYUV, 1145
 remote desktop protocol (RDP), 1143
 satisfying conditions, 1143
 screen compression and transmission test analysis, 1146
 virtual graphic processing unit, 1143
- DAEs, *see* Differential-algebraic equations (DAEs)
- Dark data
 analog-to-digital conversions, 91
 categorized and managed, 92
 collection
 dark and unstructured data, 93–94
 as a treasure, 94
 recommendations
 analytics technologies, 98
 analyzing, 96–97
 benefits, 97
 cybersecurity monitoring, 97
 risks
 into active revenues, 95–96
 data challenges, 95
 to work, 96
 user activities, 100–103
- Data aggregation, 594, 606
- Data analytics, 97
- Database-as-Service Systems, 1079
- Data collection, 71, 102, 125, 594, 606, 684–687, 842
- Data Distribution Management (DDM), 975–978
- Data generation language (DGL) software, 961
- Data mining, 1196
- Data processing, 176, 180, 319, 483, 547, 594, 605, 626, 865
- Data transmission
 fixed data transmission, 504, 599–601, 604, 605
 level data transmission, 601–605
 nodes and edges, 596
 shortest-path tree, 597
 timeline diagram, 596, 597
- Debugging process, 794
- Decentralized applications, 119, 122, 124, 338, 633, 634, 636, 640, 955, 1200
- Deep convolutional neural networks (DCNNs), 769, 770
- Deep learning
 active learning, 611, 613–615
 CNNs, 613
 drawback, 615
- Deep neural networks (DNNs), 77, 78, 80, 85–87
- Dementia, 593
- Deployed vehicles, 152–154, 156, 157, 159, 162–164
- De Rijk method, 718
- Device base class
 existing design, 374–375
 new design, 377–379
- Device model class
 existing design, 375
 new design, 379–381
- DEVS model, 36, 37, 44, 45
- DGs, *see* Distributed generations (DGs)
- Dielectric polymer genome
 computational challenge, 52
 computational framework and dataset generation, 53
 high-performance, 51
 ML models, 54–56
 predictive accuracy, 56–61
 training dataset, 56
 valance-aware ReaxPQ (ReaxPQ-v) model, 53
- Differential-algebraic equations (DAEs)
 AJ matrices vs. FDJA, 384
 dynamic simulation, 406
 model/solver interface leveraging, 372
 solver interface
 device base class, 377–379
 device model class, 379–381
 stability assessment, 371
- Digital business strategy
 business opportunities, 945
 characterization of crisis

- demand-side characteristics, 948–950
 - macro level changes, 947–948
 - supply-side characteristics, 950
 - corporate strategic areas, 952
 - corporate transformation initiatives, 953–956
 - corporation, characterization of, 953
 - COVID-19, impact of, 944
 - developing models for, 956
 - digital disruption, 944, 945
 - digital transformation, 945
 - dynamics relevant, characterizations of, 951–956
 - intelligence framework, 946
 - stage of crisis handling, 952–953
 - Digital camera, 97, 364
 - Digital disruption, 945
 - Digital initiatives, 954
 - by CIO, 954–955
 - by other executives, 955–956
 - Digital marketplaces, 945
 - Digital rights management scheme (DRM), 1179
 - Direct dynamics simulation, 217
 - Discrete-time version, 816–817
 - Distributed computing, 437, 452, 453, 619, 670, 673, 675, 738, 1054, 1075
 - Distributed denial of service (DDoS), 43, 47, 317
 - Distributed generations (DGs)
 - compensators, 235, 237
 - economic index, 239–240
 - environmental index, 240
 - load model of network, 237–238
 - MT, 236
 - multi-objective whale optimization algorithm, 236
 - numerical results, 240–244
 - objective functions, 238
 - optimization method, 240
 - photovoltaic, 236
 - technical index, 238–239
 - WT, 236
 - Django (web application framework), 73, 689, 691
 - DNNs, *see* Deep neural networks (DNNs)
 - Docker
 - container runtime, 1066
 - Dockerfile, 1091
 - image, 1059
 - vs.* Singularity, 13
 - 9-DOF inertial measurement unit, 896
 - Domain knowledge (DK), 65
 - Drones, 633–638, 640, 642–644
 - Drug discovery
 - CADD use, 747, 748
 - and materials informatics, 759–761
 - research, 747
 - Dynamic simulation
 - directive-based hybrid parallel power system, 405–415
 - GridPACK™, 371–386
 - Dynamic trust-based access control (DTBAC) model, 1121
- E**
- Early-modern Japanese printed books, 683, 686, 687, 694
 - See also* Japanese character recognition
 - E-business, 130
 - Economic crisis, COVID-19, 943
 - Economic index, 239–240, 243, 244
 - Edge computing, 116, 541–543, 620, 622, 629
 - cloud and IoT layer, 629
 - distributed computing system, 619
 - IoT integration, 624
 - Edge exchange heuristic, 353–356
 - Edge server (ES), 542–554, 586
 - E-learning method, 827
 - Electromyography (EMG) sensors, 896
 - Elemental analysis
 - art works, 363
 - composition of paints, 364
 - digital camera, 364
 - experiment results, 365–367
 - image histogram, 365
 - pigments, 363
 - sample preparation, 364–365
 - Embedded system debugging experiment
 - configuration, 797–798
 - limitation, 806
 - mathematical division method, 801–804
 - partial analysis, 801–805
 - procedure, 798–799
 - recorded data, 798
 - teaching contents, for education field, 805
 - EMG-based interaction, 897–898
 - EMG-based interfaces, 906
 - Empirical data, secBIRpts
 - business documents correctness, 1114–1115
 - data visualization correctness, 1113–1114
 - email correctness, 1115–1116
 - privacy, 1116

- Enterprise backend as a service (EBaaS)
 - Airbnb, 1078
 - application, 1095
 - application design, 1079
 - architecture, 1080–1081
 - backend-as-a-service, 1079
 - back-end developers, 1078
 - cloud services, 1097
 - code quality, 1096
 - database connect, 1096–1097
 - database system, 1082
 - data scientists, 1078
 - deployment, 1090–1092
 - evaluation methodology, 1093–1094
 - file structure comparison, 1097
 - front-end developers, 1078
 - generating code
 - architectural styles, 1086
 - database connection, 1088, 1089
 - file structure, 1087, 1088
 - mapping models, 1088
 - RESTful architectural style, 1086
 - URIs, 1089, 1090
 - generation of models
 - ALTER TABLE, 1086
 - creation, database form, 1084
 - database system, 1083
 - user interface, 1082, 1084, 1085
 - maintenance, 1092
 - methodology
 - database system, 1082
 - processes, 1081
 - model-driven approach, 1080
 - nontechnical people, 1079
 - problem statement, 1080
 - REST and RESTful, 1077
 - usability, 1094
 - user, 1081
 - user interface (UI) design, 1079
 - web application, 1077
 - web services, 1077
- Environmental index, 240, 244
- ES, *see* Edge server (ES)
- ESA Messenger benchmark, 671, 677, 686
- e-Science approach, 1196
- The European Space Agency (ESA), 669, 670, 679
- Even-dimensional bicube, 537–538
- Expressing business goals, SURF technology
 - communication price, 1163
 - inter-cloud networks, 1164
 - replication target conversion table, 1165
 - security target conversion table, 1164
 - urgency target conversion table, 1165
- Extensible Markup Language (XML)
 - document, 1102
- Eye gaze data
 - acquisition and analysis of, 794–795
- F**
- Factorization-based methods, 276
- Fake review prevention, 119
- False-position method, 716, 717
- Fiber-reinforced plastic (FRP), 201
- Field of view (FOV), 896
- Field-programmable gate array (FPGA), 481–483
 - encryption, 469
 - experimental results and analysis
 - comparisons and analysis, 473–474
 - experimental setup, 473
 - FPGA-GPU-CPU heterogeneous system, 477–491
 - heterogeneous NTRUEncrypt platform
 - memory hierarchy, 466–467
 - overview, 465
 - programming model, 466
 - Kernel optimization, 469–472
 - key generation, 468
 - NTRUEncrypt, 462–464
- Fixed data transmission, 504, 599–601, 604, 605
- Flexibility, 3, 83, 126, 375, 384, 477, 481, 557, 827, 831, 876, 896, 1079
- Flood attack, 43–47
- FMUBCT model
 - abnormal detection rate, 1133
 - authentication module, 1122
 - authorization module, 1122
 - behavior monitoring module, 1122
 - comparison module, 1123
 - flowchart, 1123
 - fuzzy logic module, 1123
 - COG method, 1129
 - comprehensive trust value, 1128
 - rapid decrease and slow-rise strategies, 1132
 - trust computation module
 - direct trust, 1124–1126
 - history trust, 1126
 - indirect trust, 1126–1127
 - trust dynamic management module, 1124
 - trust types, 1132, 1133
 - user profiling module, 1123
- Forecasting, 845
- FPGA, *see* Field-programmable gate array (FPGA)

- FPGA-GPU-CPU heterogeneous system
 architecture, 477
 FPGA OpenCL and GPU CUDA
 programming model, 481
 hardware
 architecture, 479
 logic design, 477
 heterogeneous
 kernels, 486–487
 programming, 479, 480
 high-level architecture, 478
 implementation and results analysis,
 487–491
 Nvidia GPU, 479
 oneAPI programming model, 477
 performance and energy consumption, 479
 programming model
 Intel FPGA SDK for OpenCL, 484–485
 Nvidia nvcc for CUDA, 485–486
 system architectures
 FPGA, 481–483
 GPU, 483–484
 Front-end processing, 686, 690
 FRP, *see* Fiber-reinforced plastic (FRP)
 Fused multiply-accumulate, 714, 715, 717
- G**
- GA, *see* Genetic algorithms (GA)
 Games, 659
 Game theory, 579, 581
 Gantry robots
 FRP, 201
 structural stability simulation, 206–211
 three-axes Cartesian, 202–206
 vibration reduction verification, 211–214
 welding, 201
 Gaussian Graphical Model (GGM), 781
 Gaussian process-hidden semi-Markov model
 (GP-HSMM), 794
 Gaze-fixation data, 800
 Gaze-object data, 799–801
 Gene Expression Omnibus1 (GEO), 1196,
 1197
 General-purpose computing
 in AI/ML, 494
 CPUs, 418, 420
 FPGA, 477, 478
 on GPU, 406
 hardware architectures, 495
 Programming Nvidia GPU, 483
 Genetic algorithms (GA), 542, 620, 623,
 627–628
 Genomic Expression Archive (GEA), 1196
 Georgia Gwinnett College (GGC)
 conceptual system, 862
 elements/subsystems, 861
 outside elements/subsystems, 861
 Givens rotation
 AIRLB algorithm, 707
 generalized, 704
 for higher accuracy, 714, 715
 correction of $\cos(\theta)$, 717
 false-position method, 716
 fused multiply-accumulate, 717
 secant method, 716–717
 improvements in proposed implementation,
 718
 mrTIGO scheme, 278
 OQDS algorithm, 706
 use, 278
 Global positioning system (GPS), 579
 Global trajectory optimization problems
 (GTOP), 669, 670, 678
 GNC network, *see* Greedy Navigational Core
 (GNC) network
 GNNs, *see* Graph neural networks (GNNs)
 Goal program, SURF technology, 1150
 deviation variables, 1155
 dynamic data correlation desirability, 1160
 estimate communication cost, 1167–1171
 external system description parameters,
 1157, 1158
 formulation, 1165–1167
 notation, 1155–1156
 operational statistics, 1157, 1159
 optimization directives, 1157, 1159
 performance statistics, 1157, 1158
 QCQP problem, 1155
 static desirability, 1160
 system directives, 1157, 1158
 system model, 1156–1157
 target translation tables, 1171
 utilization of each channel, 1162
 Golub–Kahan–Lanczos (GKL) algorithm, 698
 Google’s Container Engine, 1075
 GPU, *see* Graphics processing unit (GPU)
 GPU computing, 406–407, 409, 415, 487,
 742–744
 Graphical Lasso, 781, 782
 Graphical user interface (GUI), 832
 multiple learning panels, 936–938
 runtime interface, 934–935
 supply and inventory, 936
 three-level control panel, 935–936
 Graphics card, 419, 420, 423, 425

- Graphics processing unit (GPU), 483–484
 - background, 418–420
 - experimental results, 426–431
 - Faugère’s F4 and F5 algorithms, 418
 - FPGA-GPU (*see* FPGA-GPU-CPU heterogeneous system)
 - implementation, 422–426
 - methodology, 420–422
 - multivariate polynomials, 417
 - NVIDIA GPU, 418
 - S-polynomials, 417
- Graph neural networks (GNNs), 52
- Greedy Frames, 342, 343
- Greedy navigation
 - average networks, 341–342
 - discussion, 343–344
 - distributed operation, 337
 - GNC, 338
 - Greedy Frames, 342, 343
 - hop-by-hop routing strategy, 337
 - individual networks, 339–341
 - methods, 345
 - real networks, 338
- Greedy Navigational Core (GNC) network
 - average networks, 341–342
 - discussion, 343–344
 - distributed operation, 337
 - Greedy Frames, 342, 343
 - individual networks, 339–341
 - methods, 345
 - real networks, 338
- GridPACK™
 - DAEs, 371
 - DAE solver interface
 - device base class, 377–379
 - device model class, 379–381
 - dynamic simulation module, 371
 - first-order differential equations, 372
 - software framework
 - device base class, 374–375
 - device model class, 375
 - validation and preliminary results
 - accuracy, 382–383
 - flexibility, 384
 - performance, 384–385
 - variable step, 372
- Gröbner bases
 - background, 418–420
 - experimental results, 426–431
 - Faugère’s F4 and F5 algorithms, 418
 - implementation, 422–426
 - methodology, 420–422
 - multivariate polynomials, 417
 - NVIDIA GPU, 418
 - S-polynomials, 417
- Ground Contact Element (GCE) model, 876
- H**
- Healthcare, 558, 577, 594, 606, 623, 927, 928, 951, 1131
- Heterogeneous combat network
 - communication network, 992
 - coordination and cooperation, 991
 - design of repair model
 - cost constraint, 996
 - operational capability, 996
 - optional edge constraint, 996
 - optional edges, 995
 - effectiveness analysis, 999–1000
 - functional reliability, 993–994
 - modeling, 992–993
 - network damage, 991
 - network sizes, 1001–1002
 - nodes and edges, 991, 992
 - performance comparison, 1000–1001
 - repair algorithms, 1000–1001
 - solution of repair model
 - fitness function, 997
 - parameter coding, 997
 - population initialization, 997
 - simulation environment settings, 998–999
 - topological theory, 991
 - weapons and equipment systems, 991
- Heterogeneous environment
 - characteristics of, 996
 - cluster application, 453
 - computing
 - infrastructures, 479
 - system, 464
 - CPU-FPGA, 464
 - CUDA and OpenCL, 480
 - FPGA-based heterogeneous system (*see* Field-programmable gate array (FPGA))
 - FPGA-GPU, 486–487
 - heterogeneity issues, 453–454
 - modeling and simulation, 882
 - NTRUEncrypt platform
 - memory hierarchy, 466–467
 - overview, 465
 - programming model, 466
 - prototype, 664
- Hidden Markov model (HMM), 795–796
- Hierarchical Dirichlet process (HDP-GP-HSMM), 796
- High-degree first algorithm (HDF), 1000

- Higher education institutions (HEI), 859
 - Georgia Gwinnett College (GGC)
 - conceptual system, 862
 - elements/subsystems, 861
 - outside elements/subsystems, 861
 - local community and, 862–863
 - High-Level Architecture (HLA)
 - DDM, 975–978
 - distributed simulation, 976–978
 - “Hasty Attack” scenario, 983–984
 - individual simulation model, 975
 - matching, 976
 - matching intervals, 984, 985
 - numeric values, 987
 - research question and procedure, 979–980
 - RSAF, 980–983, 988
 - SAF systems, 978–979
 - sensors, 975
 - with shooting, 984, 986
 - vehicles, 984
 - without shooting, 984, 985
 - High-performance computing (HPC), 387, 405
 - awareness, 738
 - coding effort, 410
 - container images, 13
 - containerization, 11–14
 - deployed, 478
 - dielectric polymers, 51
 - high-level dynamic languages, 3
 - HPC-based applications, 405
 - matrix calculations, 3
 - and MPI applications, 13
 - MXHPC, 673
 - OS, 4
 - power
 - system simulators, 406
 - utilities, 405
 - in power grid, 406–407
 - power grid simulation, 371, 406–407
 - for protein–protein docking, 737–744
 - Raspberry Pi, 452
 - RTMaps, 80–81
 - Singularity, 4
 - High-throughput gene expression data, 1195
 - Holographic applications, 750, 751
 - HoloLens
 - basic specification, 749
 - images, 753
 - Microsoft HoloLens, 748–750, 755
 - unity, 750
 - UWP, 750, 751
 - as visualization device, 748
 - VR and AR devices, 749
 - HoloMol system
 - development environment, 750
 - evaluation, mesh binding, 753–755
 - frames drawn per second (FPS), 753, 754
 - holographic applications, 751
 - HoloLens, 751–754
 - object integration and mesh binding, 752
 - PDB, 752
 - system overview, 751
 - visualization, 751
 - HoloToolkit, 899, 904
 - Homomorphic encryption, 1178, 1179
 - Householder transformation, 713, 714
 - HPC, *see* High-performance computing (HPC)
 - Human-computer interaction (HCI) concepts, 832
 - Hybrid parallel power system
 - commercial simulation tools, 405
 - CPU-based computing, 406
 - dynamic simulation, 407–408
 - future work, 415
 - HPC in power grid, 406–407
 - proposed approach
 - algorithm decoupling, 408–409
 - parallel implementation, 409–411
 - system configuration, 411
 - results and analysis
 - OpenMP-only implementation, 414
 - scalability performance, 412–413
 - testing cases, 412
 - transient stability analysis, 407–408
 - Hyperparameter selection, 786
 - Hyper Text Transfer Protocol (HTTP), 1078
- ## I
- IaaS, *see* Infrastructure as a service (IaaS)
 - Identification pattern, 397–400
 - Image histograms, 365–367
 - Image processing, 80, 327, 364, 433, 436, 453, 613, 615, 769
 - Improved decision support system (IDSS), 842
 - Inactive neuron, 70
 - Inertial measurement units (IMU), 896
 - Infrastructure as a service (IaaS)
 - cloud computing technology, 619
 - cybersecurity, 308
 - data center locality, 316
 - encryption
 - at rest, 317–318
 - in transit, 318–319
 - problem statement and discussion
 - agile programming, 314–315
 - security support, 315–316
 - security training for employees, 315

- Infrastructure as a service (IaaS) (*cont.*)
 - recommendations
 - CIS, 322–323
 - cloud security comparison data, 321–322
 - multiple cloud providers, 320
 - security of the cloud, 321
 - related work, 308–309
 - security threats, 307, 309–314
 - top three cloud platforms, 319
 - uptime/availability of platform, 316–317
- Inheritance concepts, 586
- Integrated development environment (IDE), 831
- Integration scheme, 218, 371, 372, 374, 385
- Intel FPGA SDK for OpenCL, 484–485
- Intelligence framework, 946
- Interaction model
 - design and implementation
 - advantages and limitations, 902
 - client-server architecture, 900–901
 - design considerations, 898–899
 - free interaction task, 903
 - gesture evaluation, 903
 - HoloToolkit, 899
 - introduction and learning, 903
 - Myo-based interaction model, 900
 - EMG-based interaction, 897–898
 - freehand interaction in mixed reality, 897
- Interactive games, 928
- Interactive objects, 933
- Intercommunication, 493, 494, 502
- Interconnection networks, 451, 525, 527–530
- Internal Model Control (IMC) scheme, 812
- Internet of things (IoTs), 453, 793
 - architecture, 624–626
 - challenges, 106, 107
 - connectivity, 109
 - data storage, 109–110
 - limitations, 108
 - maintenance, 110
 - network, 110
 - privacy, 107–108
 - requirements, 109
 - security, 108
 - cloud-edge centric architecture (*see* Cloud-edge centric IoT architecture)
 - computer environments, 594, 624
 - diagnostics and status updates, 92
 - expert's technical tool, 105
 - IaaS, 619, 620
 - management, 106
 - recommendations
 - add security, 111–112
 - cloud and gateway architectures, 112
 - monitoring, 111
 - risk assessment, 111
 - secure hardware, 113–115
 - segmenting networks, 112–113
 - in smart city (*see* Smart city)
 - uncertainty and business risk, 105
 - VLANs, 116–117
 - VPN, 115–116
- Internet of vehicles (IoVs)
 - coverage model, 544–545
 - edge computing, 542
 - ESs, 542
 - experimental evaluation
 - average waiting time, 554
 - coverage rate comparison, 551–552
 - workload variance, 552–553
 - load variance model, 545–546
 - network model, 543–544
 - NPGA-II, 548–551
 - problem formulation, 547
 - related work, 542–543
 - RSUs, 541
 - waiting time model, 546–547
- Interoperability, 112, 415, 623, 639, 976, 977, 1110
- Inter-process communication (IPC), 1075
- Invariant statistical embedding and averaging
 - in terms of pivotal quantities (ISE&APQ), 257
 - preliminaries, 258–260
 - quantile estimation, 261–264
 - shortest-length confidence intervals, 266–268
 - statistical decision rules, 261
 - techniques, 257, 260–261
 - uniformly non-dominated decision rule, 264–266
 - within-sample prediction limits, 268–273
- Inventory control, 842
 - ABC analysis, 843–845
 - forecasting, 845
- Inventory cost, 842
- Inventory monitoring, 839
- Inventory system
 - analytic hierarchy process (AHP), 842
 - demand distribution function, 840
 - identification of items, 839
 - improved decision support system (IDSS), 842
 - inventory control costs, 840
 - optimal inventory model, 841
 - optimal inventory policy, 840
 - procurement and supply, 840

- scoring, 840
- selection, 840
- service level analysis, 840
- simulation-based optimization method, 841
 - Vensim software, 841
- Inventory-to-sales ratio, 841
- IoT, *see* Internet of things (IoTs)
- IoV, *see* Internet of vehicles (IoVs)
- iPython parallels, 1075
- ISE&APQ, *see* Invariant statistical embedding
 - and averaging in terms of pivotal quantities (ISE&APQ)
- Island genetic algorithm (GA)
 - BDL method, 726–727, 729–731
 - conventional, 725, 733, 734
 - with different fitness functions, 728–731
 - environmental differences, 725
 - experiments, 731–733
 - initial and final generations, 732, 734, 735
 - multi-objective optimization, 725
 - packing problems, 726–727
- J**
- Jacobi method, 715
 - computation accuracy and speed, 714
 - Givens rotation with high accuracy, 722
 - Jacobi rotation, 715
 - off-diagonal components zero, 718
 - one-sided, 714, 721–723
 - and QR method, 714
- Japanese character recognition
 - appearance frequency, character types, 687–688
 - black pixel ratio, character images, 689, 690, 692–693
 - character recognition process, 690
 - collecting characters by crawler, 688–689
 - early-modern printed books, 683
 - high and low appearance frequency, 689
 - implementation for character images
 - selection, 689–691
 - JIS level 1 and 2, 684
 - practical early-modern, 684
 - research, 683
 - web application for learning data collection, 684–687
 - Zipf's law, 684, 687, 688
- JIT compilation, *see* Just-in-time (JIT) compilation
- Julia programming language
 - JIT, 5–8
 - LLVM, 8
 - native GPU support, 8–11
 - Just-in-time (JIT) compilation, 5–8
- K**
- Kanji characters, 689
- KB controller design, 924–925
- KB parameterization, 914
 - closed control loop, 917
 - reference signal tracking, 918
 - transfer characteristics, 917
- Krylov subspace method, 698, 701, 707
- L**
- Large-scale parallelization
 - experiment environments, 144
 - experiments, results and analysis, 145–146
- Lattice quantum chromodynamics (LQCD)
 - calculation theory, 136–139
 - high-energy physics theory, 134
 - high-precision calculation, 133
 - innovations realized
 - algorithms, 143–144
 - CPEs data allocation, 139–141
 - multiple core groups, 141–142
 - parallelization method, 141–142
 - pseudocodes, 143–144
 - overview, 136
 - SW26010 heterogeneous many-core processor, 134–136
 - theory, 133
- Learning data
 - character recognition accuracy, 685
 - collection method by crawling, 688–689
 - for early-modern Japanese printed character recognition, 683–688
 - redevelopment, web application, 686
- Learning methods, 85, 612, 615, 769, 827
- Learning models, 54–56, 218, 225, 623, 760, 761, 897, 946, 956
- Least absolute shrinkage and selection operator (LASSO), 781
- Leave-one-element-out cross-validation, 761, 765, 766
- Lifestyle diseases, 593
- Ligand-based drug design (LBDD), 748
- Linear Algebra PACKage (LAPACK), 410, 706, 714, 721, 724
- Linear least squares
 - direct and iterative methods, 275–276
 - factorization-based methods, 276
 - generic approximate sparse pseudoinverse scheme, 277–278
 - gradient method, 276–277

- Linear least squares (*cont.*)
 numerical results
 computed error estimates, 281–289
 computed relative error, 281–289
 model problems, 280–281
 preconditioning methods, 276–277
 sparse coefficient matrix, 275
 theoretical estimates, 279–290
- Line thickness, 328, 329, 333–334
- Lithography simulation modeling, 697, 698, 707, 711
- LLVM, *see* Low-level virtual machine (LLVM)
- Load data, 143, 393, 1200
- Load model
 of network, 237–238
 voltage-frequency and various customers, 236
- Long short-term memory (LSTM), 220–222
 deep learning, 218
 dynamics simulations, 218
 experimental results, 225–232
 Hamiltonian methods, 217
 model, 222–225
 numerical integration, 217
 PES, 217, 218
 prediction-correction algorithm, 219–220
- Loss prediction module, 614
- Low betweenness first (LBF), 1000
- Low degree first (LDF), 1000
- Low-level virtual machine (LLVM), 5, 6, 8, 9, 500, 502
- LQCD, *see* Lattice quantum chromodynamics (LQCD)
- LSTM, *see* Long short-term memory (LSTM)
- M**
- Machine learning (ML)
 active learning, 609
 band gap prediction model, 766
 computational compound enumeration, 759
 deep learning, 218
 democratized ML tasks, 52
 leave-one-element-out cross-validation, 761, 765, 766
 methods, 761–763
 models, 54–56
 molecular
 and materials sciences, 52
 property prediction, 759
 molecular property prediction, 759
 polarizable reactive force-field model, 61
 predictive accuracy, 56–61
 QM, 218
 schematic representation, 56, 57
 SD model, 956
 selected features, 761
 services, 1081, 1098
 smart surveillance system, 623
 tools, 70
- Malware detection
 agent-based modeling, 581
 keystroke logging, 311
 ransomware, 32
 target group/organization, 36
 Trojan, 17, 19, 23
- Mandatory Access Control (MAC), 1101
- Markov chain, 795–796
- Markov chain Monte Carlo method, 785
- Markov model, 802
- Materials informatics, 759–761
- Mathematical division method, 801–804
- MATLAB software, 851, 880, 881
- Maximum a posteriori (MAP), 782
- MEGADOCK (protein–protein docking tool)
 on Azure CPU, 741
 FFT calculation, 740
 multi-node implementation, 740
 on multiple CPU instances, 741–744
 on multiple GPU instances, 742–744
 protein–protein interaction prediction, 740
 public cloud environment, 744
 RDMA network, 743
 supercomputer-powered software, 739
- Memory retention, 931
- MEMS Gyroscope
 AC response analysis, 883
 amplitude and phase for frequency, 885
 box beam springs application, 883
 different inertial mass dimensions, 884
 modal analysis results, 884
 simulink model of, 882
 transient analysis, 884, 885
- Mercury, 875–876
- Mesh binding, 752–755
- Message passing interface (MPI)
 hardware and benchmarks details, 454–455
 heterogeneity issues, 453–454
 message passing, 451
 parallel computing, 451
 performance results, 455–457
 Raspberry Pi models, 452
 related work, 452–453
 standard computing clusters, 452
- Messenger (full mission) benchmark in GTOP
 database
 GTOP benchmark, 670

- multi-gravity assist interplanetary space mission, 671
 - MXHPC/MIDACO algorithm, 673–675
 - newly presented results vs. previously published ones in 2017, 676–679
 - numerical results, MXHPC, 675–676
 - as optimization problem, 669
 - optimization variables, 671
 - published numerical results, 672–673
 - Metropolis-Hastings sampling scheme, 785, 786
 - Microsoft Azure
 - computing environment, 739
 - HPC, 738
 - MEGADOCK software, 741, 744
 - public clouds, 737
 - RDMA, 738
 - Microsoft HoloLens, 748–750, 755
 - MIDACO Extension for High-Performance Computing (MXHPC) algorithm, 670, 671, 673–679
 - Million instruction per second (MIPS), 628
 - Minimization optimization procedure, 891
 - ML, *see* Machine learning (ML)
 - Mobile ad hoc networks (MANET)
 - error recovery, 660
 - fundamental characteristics, 659
 - implementation, 664
 - node, network functions, 659
 - with ring topology (*see* Ring topology, MANET)
 - tests, 663–664
 - Mobile cloud computing (MCC)
 - call chart
 - CPU profiler, 1185
 - view card activity, 1186
 - CPU profiler, 1184, 1185
 - DRM framework, 1179
 - homomorphic encryption, 1178, 1179
 - literature review, 1178–1180
 - Paillier and AES compression, 1186
 - proposed system
 - application interface, 1184
 - encrypted form, 1184
 - initialization and authentication, 1180–1181
 - registration page, 1183
 - secure data, 1182
 - software model, 1183
 - storing data, 1181–1182
 - symmetric homomorphic encryption, 1182
 - system design, 1181
 - user identity in cloud, 1183
 - strengthening the authentication process, 1179
 - system performance, 1187
 - Model-based youla regulator, 919–920
 - Model composition, PNPSC, 586
 - Model-Driven Web Engineering, 1079
 - Model error, 921
 - Modeling
 - and assessing, 67
 - deployed vehicle, 157
 - knowledge, 154
 - phase, 68
 - and real-time analysis, 97
 - threat, 152
 - Modeling and simulation (M&S), 887
 - Modified Gram–Schmidt algorithm, 702, 703, 708
 - Molecular rift, 748
 - Monte Carlo method, 137, 167, 171, 614, 615, 785
 - MPI, *see* Message passing interface (MPI)
 - Multi-cloud, 1055
 - Multicommodity flow formulation, 351–352
 - Multicore
 - CPU (*see* Central processing units (CPU))
 - GPU, 170
 - and many-core compute servers, 168, 172
 - MKL library, 175
 - PRNG-Broker, 181
 - processors, 80
 - Multi-objective optimization, 240, 543, 547, 549, 725
 - Multivariate polynomials, 417
 - Myo-supported gestures, 906
- N**
- National League for Nursing (NLN), 825
 - National Science Foundation, 1059
 - Neural network
 - GNNs and RNNs, 52
 - Merck’s internal baseline model, 218
 - structure, 221
 - structure-property relationships, 52
 - Neurological system module, 837
 - Newton-Raphson method, 785
 - Non-verbal techniques, 794, 796, 801, 806
 - Normalization, 81, 771
 - NoSQL database
 - big data, 558
 - data management tools, 557
 - experiments and comparison
 - comparison, 572, 575
 - technical environment, 571–572

- NoSQL database (*cont.*)
 ToConceptualModel process, 572–574
 validation, 575–577
 flexibility, 557
 future work, 577
 illustrative example, 558–559
 related work, 559–560
 reverse engineering process, 560–571
- NPGA-II
 edge server placement scheme, 548–551
 encoding strategy, 548
 workload variance, 552–553
- NTRUEncrypt
 decryption, 464
 encryption, 464
 heterogeneous platform
 memory hierarchy, 466–467
 overview, 465
 programming model, 466
 key generation, 463–464
 notation, 462–463
- Nursing education
 game design, 929–930
 game implementation
 evaluation system, 938–939
 graphical user interface (GUI), 934–938
 interaction and animation, 933–934
 sustained learning, 940
 3D hospital environment, 931–933
 interactive games, 928
 simulation tool and distance education, 928
- NVIDIA GPU, 8, 13, 173, 418, 419, 479, 483, 486, 652
- Nvidia nvcc for CUDA, 485–486
- O**
- Object oriented programming (OOP), 373, 436, 582, 831
- Object relational mapping (ORM), 1088
- Odd-dimensional bicube, 531–537
- Oil paints, 364
- One-sided Jacobi method
 conventional implementation, 714
 Jacobi rotation, 721
 pseudocode of proposed implementation, 719
- Online sales, 949
- Online Transaction Processing (OTP) systems, 1101–1103
- OpenMP, *see* Open multi-processing (OpenMP)
- Open multi-processing (OpenMP), 170, 249, 406, 410–415, 486, 740
- Operational initiatives, 954
- Orientation determination, 328–333
- Orthogonalization computation, 718
- Orthogonal-qd-with-shift (OQDS) algorithm, 698, 701, 703–704, 706, 707, 711, 713, 721–723
- P**
- Paint analysis
 art works, 363
 composition of paints, 364
 digital camera, 364
 experiment results, 365–367
 image histogram, 365
 pigments, 363
 sample preparation, 364–365
- Paint pigments, 364
- Pangaea, 861
- Parallel analysis module, 389, 393, 397
- Parallel analyzer, 388, 392, 396, 400, 401
- Parallelization
 ATMC algorithm, 248
 atomistic simulations, 248
 F4/5 algorithm, 431
 lattice QCD calculation, 134, 136
 on multiple core groups, 141–142
 MXHPC algorithm, 670, 673, 676
 NTRUEncrypt, 462
 PPy condensed phase, 249–252
 PRNG strategies, 172, 175
- Parallel processing
 C++, 436
 dividing tasks, 433
 hardware components, 435
 image and video processing, 436
 motivation, 437
 multiple connected processors, 434
 OpenCV, 436
 Python implementation, 436
 Raspberry Pi RISC, 433, 435
 serial graphics rendering, 448
 methods, 444–445
 results, 444, 446
 serial password hashing
 futurework, 447
 methods, 438, 440–442
 results, 440, 443
 serial sorting
 futurework, 447
 methods, 437–439
 results, 438
- Parallel programming, 433, 458, 466, 494, 497, 498, 504, 505

- Parameterizations
 - KB controller design, 924–925
 - KB parameterization, 914
 - closed control loop, 917
 - reference signal tracking, 918
 - transfer characteristics, 917
 - observer principle, 920–923
 - Youla parameterization
 - closed-loop control systems, 915
 - control loop, 913
 - equivalent IMC loop, 915
 - internal model control, 914
 - transfer function, 912
 - two-degree-of-freedom systems (TDOF), 914
 - Youla-Parameterized controller design, 924
- Parametric uncertainty, 257, 273
- Parent–child relationships, 580–582
 - CAPEC-reported attack patterns, 582–583, 586
 - composition example, 587–589
 - PNPSC models, 583–586
 - relationship composition, 587
- Pareto analysis, 946
- Pareto chart, 844
- Pareto’s analysis method, 843
- Parkinson’s disease, 835
- Partial correlations, 788–789
- Particle mesh Ewald (PME), 1024
- Particle Swarm Optimization, 889, 890
- PAS2P tool, 388, 390–392, 400, 401
- Pathophysiology
 - neurological system, 835, 837
 - pulmonary system, 835, 836
- Pattern identification, 389, 393, 397, 398
- Pb-free perovskites, 760, 766, 767
- Perovskite solar cells, 760, 766, 767
- PES, *see* Potential energy surface (PES)
- Petri Nets
 - composability, 580
 - definitions, 580
 - with PNPSC, 580
 - web services control flow, 579, 580
- Petri Nets with Players, Strategies and Cost (PNPSC), 580–586
- Photo-trapping, 659
- PlainNet, 770
- Platform as a Service (PaaS), 619, 1119, 1178
- Poisson process, 782, 791
- Policy changes, 951
- Poly(lactic-co-glycolic acid) (PLGA), 1023
- Polygon model, 829
- Polymers
 - antibiotics and macromolecules, 1023
 - atomic species, 53
 - chemical and morphological features, 52
 - dielectric constants, 56
 - functional groups, 56
 - high-performance dielectric, 51
 - PPy (*see* Polypyrrole (PPy))
 - workflow of tasks, 1025
- Polypyrrole (PPy)
 - condensed phase analysis, 249–252
 - heterocyclic aromatic monomers, 247
 - methods, 248
 - oxidized phase, 247
 - parallelization protocol, 249–252
 - thermodynamic properties, 252–253
- Pool-based active learning, 610–611
- Porter’s five forces analysis, 946
- POSIX threads (pthreads), 498
- POST-DS (Process Organization and Scheduling electing Tools for Data Science)
 - data mining, 65
 - data science model, 70–71
 - knowledge discovery, 65
 - literature review, 66–70
 - using the model, 71–73
- Potential energy surface (PES), 217–219
- PPy, *see* Polypyrrole (PPy)
- Preconditioning technique, 698–701
- Pressure couplings, 1024
- Privacy, 107–108
 - business intelligence platform, 1102
 - cloud computing, 1179
 - data, 1110–1112
 - empirical data, 1116
 - ethics and, 69
 - physical distance, 949
 - security and, 105, 315
- PRNG, *see* Pseudorandom number generation (PRNG)
- PRNG-Broker
 - API, 174–175
 - case studies, 177–178
 - core of the broker, 172–173
 - hardware-aware PRN generation, 173–174
 - libraries, 174–175
 - Mersenne Twister PRNG, 168
 - server configurations, 179
- Probability density function (PDF), 845
- Programmable logic controller (PLC), 1018
- Protein Data Bank (PDB), 748
- Protein–protein docking, 738–744
 - See also* MEGADOCK (protein–protein docking tool)

- Pseudocodes, 22, 143–144, 444, 445, 675, 714, 718–721
- Pseudoinverse matrix, 278–280, 290
- Pseudorandom number generation (PRNG)
- associated case studies, 177
 - case studies with PRNG-Broker, 177–178
 - comparative performance, 178–180
 - digital chips, 167
 - PRNG-Broker, 171–175
 - random number generation, 168–171
 - scientific
 - application, 167
 - community, 168
 - data analyses, 175–176
 - servers, 178–180
 - test beds, 177
- Pulmonary system module, 836
- Python
- CovidLock ransomware, 32
 - functionality, 436
 - high-level dynamic language, 3
 - implementation, 439
 - interpretive language, 423
 - packages, 81
 - programming language, 18, 19
 - RTMaps, 87
 - scaling, 1067
 - for server-side programs, 689
 - simulation model, 37
 - syntax, 5
 - worker nodes, 437
- Q**
- QCD, *see* Quantum chromodynamics (QCD)
- Quantile estimation, 261–264
- Quantum chemical calculations, 760
- Quantum chromodynamics (QCD)
- lattice QCD (*see* Lattice quantum chromodynamics (LQCD))
 - theory, 133
- Query-by-committee (QBC), 612–613
- QuickAccess panel, 931
- R**
- Random addition (RA), 1000
- Random data generation
- bridge hands, 961–962
 - DGL, 961
 - experimental data
 - bicycle new deck order, 967
 - bridge rank order, 966
 - hands, random order, 967, 968
 - indexed sorting, 967, 971
 - keyed sorting, 967, 970
 - simple sorted hands, 967, 969
 - trimming, 967, 972
 - indexed sorting, 964, 965
 - keyed sorting, 964
 - simple sorting, 962–963
 - trimming, 965, 966
- Random number generation
- popular PRNG algorithms, 169
 - PRNG libraries, 170–171
 - uniformly distributed PRNs, 169–170
- Raspberry Pi
- brute force password, 443
 - C++ program, 445
 - hardware components, 435
 - language environments, 437
 - MPI, 441 (*see also* Message passing interface (MPI))
 - parallel processing, 433, 448
 - RISC, 433
- Recurrent neural networks (RNNs), 52, 54, 55, 61, 218, 220, 221
- Reduced Echelon Format (REF), 418, 421
- Regularization techniques, 697
- Remote Desktop Session Host (RDSH)
- CPU usage, 1193
 - existing analyses, 1189
 - experiment design, 1190–1191
 - Knowledge Worker v2, 1190, 1192
 - MMR application, 1192
 - multimedia redirection, 1190
 - network utilization, 1192
 - Windows Server 2008 R2 environment, 1190
- Remote direct memory access (RDMA), 738, 741–744
- Representational state transfer (REST), 1110
- Residual network (ResNet), 769
- down-sampling block, 770
 - inactive neurons, 776–777
 - PlainNet34, 771
 - preferred stimulus in, 773–774
 - receptive field, 771–772
 - training, 772–773
 - visualization, 772, 774–776
 - visualization filters, 773
- Response surface methodology (RSM), 888
- RESTful (Representational State Transfer) system, 1077
- Restrained electrostatic potential (RESP), 1024
- Reverse engineering process
- conceptual model, 562–564
 - physical model, 561–562

- target, 562–564
- transformation algorithms, 564–571
- Reverse threat modeling
 - “age”, 152
 - approach, 156–159
 - contribution, 152
 - driver assistance systems, 151
 - example, 159–163
 - problem statement, 152
 - solution, 152
 - TARA, 153
 - vulnerabilities, 151
- Reward and punishment trust model (RPTM), 1121
- Ring topology, MANET
 - algorithm, 662
 - application header desthdr, 661
 - error recovery, 660
 - login message header, 661
 - message handle algorithm, 663
 - message transmission, 662
 - network creation, 661
 - network login, 661
 - ring network topology diagram, 660
 - routing networks, 660
 - routing protocols, 660
 - routing table, 661
 - verification, networks, 662
- RISC processor, 433, 435, 437, 448
- RNNs, *see* Recurrent neural networks (RNNs)
- Roadside units (RSUs), 541–549, 551
- Robotics, 417
- Run-Time Infrastructure (RTI), 975, 977

- S**
- Sakurai–Sugiura method, 713, 714
- Scalability, 106, 139, 390, 412–413, 634, 635, 640, 641, 663, 988, 1059–1076
- Secant method, 716, 717
- secBISQL (SQL version)
 - email service provider, 1110
 - entity relationship (ER) diagram, 1108, 1109
 - to measure, 1108
 - REST data, 1110
 - SAAS word processing solution, 1108, 1109
- Secure Business Intelligence Report (secBIrpts)
 - attributes and child elements, 1107–1108
 - cloud application, 1102
 - cloud providers, 1103
 - data mining process, 1102
 - empirical data
 - business documents correctness, 1114–1115
 - data visualization correctness, 1113–1114
 - email correctness, 1115–1116
 - privacy, 1116
 - expression tags, 1106–1107
 - MAC, 1101
 - MySQL system, 1104
 - OTP data, 1102, 1103
 - RDBMS RBAC systems, 1103
 - runtime engine, 1113
 - secBI programming languages, 1102
 - secBISQL
 - email service provider, 1110
 - entity relationship (ER) diagram, 1108, 1109
 - to measure, 1108
 - REST data, 1110
 - SAAS word processing solution, 1108, 1109
 - SQL injection vulnerabilities, 1103
 - statement tags, 1104–1106
 - XML document, 1102
 - XSD file
 - ER diagram, 1111
 - privacy virtual element, 1112
- Security
 - auditing tool, 100–101
 - awareness training, 33
 - IoT (*see* Internet of things (IoTs))
 - practices and administrative privileges, 4
 - SimPy, 19
 - system’s, 19
 - See also* Cybersecurity
- Security support, 315–316
- Security threats, 108, 115, 307–314, 581, 1178
- Security training for employees, 308, 315
- Segmentation, 67, 112, 113, 116, 250, 328, 794, 795
- Semi-automated forces (SAF) systems, 978–979
- Sensitivity
 - approximate pseudoinverse matrix, 279–280
 - complementary sensitivity functions, 913
 - GASP, 276
 - of linear least squares, 276
- Sensor networks
 - data aggregation, 594
 - data collection, 594
 - health support system, 594
 - medical treatment, 594

- Sensor networks (*cont.*)
 - sensor nodes, 595
 - wearable medical sensor systems, 594
- Serial graphics rendering, 444, 446, 448
- Serial password hashing
 - futurework, 447
 - methods, 438, 440–442
 - results, 440, 443
- Serial sorting
 - futurework, 447
 - methods, 437–439
 - results, 438
- Service level analysis, 855–856
- Service provisioning, 620–622, 629
- Shipyards, 201, 202, 214
- Shortest-path routing algorithm
 - computation demands, 525
 - cube-based topologies, 527–530
 - definitions, 525–527
 - even-dimensional bicube, 537–538
 - interconnection networks, 527–530
 - odd-dimensional bicube, 531–537
 - topologies, 525
- Simplified ground vehicle suspension model, 888
- Simulation
 - APT, 36
 - attack example, 33–34
 - BYOD, 35, 36
 - comparing population, 25, 26
 - competitive gamers, 43
 - COVID-19, 25
 - DDoS, 43
 - DEVS, 37, 44, 45
 - future research, 41–42
 - hackers, 36
 - heterogeneous environment, 882
 - methodology, 32
 - analysis, 40–41
 - results, 38–40
 - network computer systems, 35
 - other mitigation methods, 32–33
 - overview, 31–32
 - PRNG (*see* Pseudorandom number generation (PRNG))
 - ransomware, 25, 26
 - ransomware in the States, 26–29
 - reaction time, 29–31
 - results, 46–47
 - server class, 45
 - structural stability simulation, 206–211
 - SYN packet, 46
 - three way handshake, 43, 44
- Simulation model, 866–867
 - extrapolation, 851–852
 - implementation, 852–853
 - MATLAB/Simulink/Cadence software, 881
 - model implementation
 - preliminary version, 869–870
 - simulation results, 870
 - students/customer arriving processes, 867–869
 - time scales process, 869
 - optimization (MATLAB), 854–855
- Simulink model, 881
- Single-commodity flow formulation, 348
- Single instruction multiple data (SIMD), 134, 169, 473, 474, 486, 488, 500, 647–650, 656
- Single program multiple data (SPMD)
 - analyzer module, 388, 389
 - experimental results, 400–403
 - multiple processors, 387
 - parallel analysis module, 389
 - parallel analyzer, 388
 - PAS2P
 - overview, 391–392
 - tool, 388
 - proposed methodology
 - application model, 393, 395–397
 - extension of parallel analysis, 393, 394
 - identification pattern, 397–400
 - load data, 393
 - related work, 390–391
 - trace analysis, 389
- Singularity
 - definition file, 4
 - docker vs., 13
 - general container architecture, 12
 - and GPUs, 13–14
- Singular value decomposition, 697, 698, 700–702, 711, 714
- Smart city, 620–621
 - features, 621
 - ICTs, 621
 - IoT devices, 620, 629
 - IoT layer, 625
 - requirement, 621
 - service provisioning, 620
 - smart healthcare, 623
 - smart security, 623
 - smart services, 622, 629
 - technologies, 622
 - types, electronic devices, 619–620
- Smart healthcare, 619, 621, 623, 625, 626
- Smith predictor, 815–816
- Sniffing attack, 587–590
- Social engineering, 18, 23, 36

- Software as a Service (SaaS), 619, 1108, 1109, 1119, 1121, 11878
- Software development life cycle (SDLC)
 - model, 830
- Solvent-accessible surface area (SASA), 1026
- Space trajectory, 669
- Sparse covariance structure analysis, 781
- SPMD, *see* Single program multiple data (SPMD)
- SqueezeNext
 - CNNs/DNNs, 77
 - deployment, 83–84
 - NXP ADAS real-time network, 77, 78
 - results, 84–87
 - shallow architecture, 81–82
- Standard computing clusters, 452
- Standard HoloToolkit interaction, 899
- State-feedback principle, 920
- State of charge (SOC), 1018, 1020
- Statistical data, 867
- Statistical decision
 - rules, 261
- Statistical decisions, 257–261, 263–265, 273
- Stochastic gradient descent method, 772
- Stream-based active learning, 610
- Strings, 962, 963
- Structural stability simulation
 - deformation distribution graph, 206, 208, 210
 - safety factor graph, 206–210, 212
 - shear force and bending moment, 207, 209
 - stress distribution graph, 206–208, 210, 211
 - structural analysis, 207, 208
 - and vibration reduction, 202
 - y, z axis shape, 210
- Structure-based drug design (SBDD), 748
- Sunway TaihuLight supercomputer
 - QCD, 134
 - software and hardware parameters, 144
 - SW26010 heterogeneous many-core processors, 134–136
- Supply side, 950
- SURF system
 - algorithm, data movement, 1036
 - assignment, 1035
 - business goal, 1034
 - characteristics, 1033
 - completely failed, 1054
 - comprehensive system, 1035
 - concurrency control, 1036
 - data and applications, 1033
 - data movement, 1039
 - data placement, 1054–1055
 - data structure, 1036–1038
 - data systems, 1035
 - decision-making process, 1036
 - distributed computing literature, 1054
 - identifying versions, 1050
 - locking implementation
 - movement signaling, 1046
 - move-mode locks, 1048–1049
 - read-mode locks, 1047
 - write-mode locks, 1047–1048
 - mathematical programming, 1034
 - mode of deployment, 1049
 - normal operation, 1051
 - objects, 1033
 - optimistic algorithm, 1041–1044
 - optimizations, 1039–1041
 - order of moving tuples, 1038
 - partial data site failure, 1053
 - partially failed, 1054
 - partial ZooKeeper failures, 1051
 - reclaiming old space, 1051
 - recovery policy, 1049–1050
 - single data system, 1035
 - system performance parameters, 1034
 - system types, 1033–1034
 - total ZooKeeper failure, 1052–1053
 - tree-based data model, 1049
 - tuple, 1053–1054
 - types of algorithms, 1034
 - ZooKeeper data structure, 1044–1046
- SURF technology
 - choice of tuples, 1152–1153
 - comprehensive system, 1150
 - consultant system, 1150
 - cost of moving, 1153–1155
 - expressing business goals
 - communication price, 1163
 - inter-cloud networks, 1164
 - replication target conversion table, 1165
 - security target conversion table, 1164
 - urgency target conversion table, 1165
 - goal program, 1150
 - deviation variables, 1155
 - dynamic data correlation desirability, 1160
 - estimate communication cost, 1167–1171
 - external system description parameters, 1157, 1158
 - formulation, 1165–1167
 - notation, 1155–1156
 - operational statistics, 1157, 1159
 - optimization directives, 1157, 1159
 - performance statistics, 1157, 1158

- SURF technology (*cont.*)
 QCQP problem, 1155
 static desirability, 1160
 system directives, 1157, 1158
 system model, 1156–1157
 target translation tables, 1171
 utilization of each channel, 1162
 hypothetical application, 1150
 measuring and evaluating performance, 1152
 problem description, 1151
 simulation system, 1171–1173
 SPANStore, 1173
 virtual machine migration, 1174
- Surrogate models, 888–889, 891
- Suspension test rig
 suspension test rig driver, 879
 suspension test rig input file, 879
- Sustainable development (SD)
 big data, 862
 education institution sustainability, 871
 GGC subject domain, 861
 higher education institutions (HEI), 859
 rational sustainable balance (RSB), 864
 structured components, 865
 TARGETS model, 861
 traditional systems classification, 865
- Swarm intelligence, 633–643
- SW26010 heterogeneous many-core processor, 134–136
- SYN flood attacks, 44–47
- System configuration, 248, 254, 411, 1024, 1068
- T**
- Tag clouds
 analysis of full design, 1009–1010
 characteristics, 1006, 1012
 design elements, 1006
 display type, 1010
 materials, 1007–1008
 participants, 1007
 procedure, 1008–1009
 psychophysics, 1006
 substantial underestimation bias, 1011
 typeface size, 1006, 1011
 visualization, 1005
- TARA, *see* Threat analysis and risk assessment (TARA)
- Task-agnostic active learning method, 614
- TCF, *see* Thick control flow (TCF)
- TCPED (tasking, collection, dissemination, exploitation and dissemination), 6443
- TCP protocol, 44
- Technical index, 238–239
- Test rig simulation
 optional input file options, 878
 tire test rig input file, 877–878
 tire test rig scene types, 878
 tire test rig tire types, 878
- Thick control flow (TCF), 494–505
- Thick control flow processor architecture (TPA), 494–497, 499–505
- Third-party auditor, 309, 312
- Thread divergence, GPU, 647–649, 654
- Threat analysis and risk assessment (TARA), 153–156, 159
- Threat identification method, 152–164
- Three-axes gantry robot
 girder shape, 202, 204
 leg shape, 202, 204
 platform shape, 202, 204
 whole shape, 202, 205
x axis shape, 202
y axis shape, 202, 203, 205
z axis shape, 202, 203, 206
- 3D model websites, 933
- 3D object manipulations, 896, 902
- 3D protein structure database, 748
- 3D Virtual Models
 application design
 delivery and testing, 832–833
 model manipulation, 829–830
 programming and integration, 830–832
 for nursing education, 826
 pedagogical strategies, 826
 virtual reality (VR) simulators, 826, 827
- Time-delay control systems
 generic two degree of freedom (G2DOF)
 closed-loop characteristics, 819
 discrete-time version, 816–817
 pulse transfer function, 820
 Smith predictor, 815–816
 for stable linear plants, 813–815
 YP-based G2DOF design procedure, 818
 transfer function, 811
 Youla parameterization, 812–813, 822
- Time-series data, 795–796
- TPA, *see* Thick control flow processor architecture (TPA)

- Traceability, 642
 - Trace analysis, 389
 - Track test rig, 879–880
 - Transmission efficiency, 147
 - Tree algorithms, 600, 603, 604, 606
 - Tree-based fixed data transmission, 594, 595, 597–601
 - See also* Data transmission
 - Trojan attack, 18
 - Trojan Banker
 - “ANIMAL”, 17
 - communication, 18
 - methodology, 18–21
 - results, 21–24
 - against victim computer, 17, 18
 - Truncated eigenvalue decomposition, 713
 - Truncated singular value decomposition, 698
 - Trust development, 641
 - Trusted AI, 635–643
 - Trusted reviews
 - architecture, 126, 128
 - blockchain technology, 130
 - model, 123–125
 - product/service, 122
 - Trust value evaluation principles
 - expiration records, 1129–1130
 - malicious behavior, 1131
 - rapid decrease strategy, 1131
 - recent behavior principle, 1130
 - slow-rise strategy, 1130
 - Typographical printing, 683
- U**
- Undergraduate research
 - Bladerunner, 293
 - FESTO PLC, 294
 - Prep 96, 293, 294
 - programming results
 - Forward 1, 301, 302
 - gantry, 303–304
 - GVL_HMI, 297
 - homogenizer machine, 300–301
 - IF statement, 305
 - parameter exit code, 297, 299
 - procedure, 296–297
 - SFC flowchart, 297, 298
 - tray profile, 300, 301
 - troubleshoot action, 303
 - variables, 299
 - software, 294–296
 - Unity (game engine), 750
 - Universal Windows Platform (UWP), 750, 751
 - Unnamed aerial vehicles (UAVs), 579
 - end-to-end trust mechanism, 638–640
 - GPS-related attacks, 579
 - mobility applications, 643
 - swarm intelligence, 633–643
 - system architecture
 - components, 636–638
 - trust factors, 635–636
 - User activities
 - auditing tool, 100–103
 - User-behavior assessment-based dynamic access control (UBADAC) model, 1122
 - User behavior modeling, 1120–1121
 - User-behavior trust model based on fuzzy logic (UBTMFL) model, 1122
 - Users’ feedback, 905–906
- V**
- Valence-aware ReaxPQ (ReaxPQ-v), 53
 - Variable step, 372, 384, 385
 - Vehicle subsystem test rigs
 - Chrono, 875
 - Ground Contact Element (GCE) model, 876
 - Mercury, 875–876
 - modeling and simulation, 874
 - physical test rig systems, 874
 - test rig class, 876
 - Vensim software, 841
 - Vibration reduction verification, 202, 211–214
 - Virtual local area networks (VLANs), 112, 115–118
 - Virtual machines (VMs), 4, 5, 620, 625, 627, 739, 1055, 1065, 1147, 1184
 - Virtual private network (VPN), 111, 112, 115–116, 954, 1140
 - Virtual reality (VR), 748, 749, 755, 826, 827, 904, 929
 - Viscous incompressible fluid flow
 - difference schemes, 187–191
 - experimental evidence, 186
 - gases and liquids, 185
 - mathematical problem, 186–187
 - step height on the flow, 195–196
 - stream
 - curves plots, 193–195
 - values, 192, 193
 - Taylor-Galerkin algorithm, 185
 - vorticity
 - curves, 19, 197–199
 - values, 192, 193
 - Vision-based systems, 895
 - Visualization, 772

VLANs, *see* Virtual local area networks (VLANs)
VPN, *see* Virtual private network (VPN)

W

Watering hole attack, 36, 37, 40–42
Wearable medical sensor systems, 594
Web applications, 582

- collecting characters by crawler, 688–689
- crawling technique, 688
- CRUD operations, 1078
- desktop-based application, 836
- Django and PostgreSQL, 689, 691
- procedure of collection work, 684
- redevelopment, 685, 686
- REST system, 1077
- security scanner, 1103

WEB-based system

- biological data mining, 1196
- e-Science approach, 1196
- GEO database, 1200, 1202
- identifier column, 1201
- partial visualization interface, 1201
- proposed system, 1197–1198
- system architecture, 1198–1200

Web services, 579
Windows-based operating systems (OS), 833
Wind turbine (WT), 236, 241, 242

Wireless sensor networks, 594
Workload variance, 542, 548, 552–553
WT, *see* Wind turbine (WT)

X

Xeon W, 493–497, 500, 502–505
XGBoost model, 762–764
XML Schema Definition (XSD) file

- ER diagram, 1111
- privacy virtual element, 1112

Y

YOLO algorithm, 799
Youla parameterization, 812–813

- closed-loop control systems, 915
- equivalent IMC loop, 915
- internal model control, 914
- transfer function, 912
- two-degree-of-freedom systems (TDOF), 914

 Youla-parameterized control loop, 913
Youla-Parameterized controller design, 924
YP-based G2DOF design procedure, 818

Z

Zipf's law, 684, 687, 688