Advances in Intelligent Systems and Computing 676

Mohammad S. Obaidat
Tuncer Ören
Yuri Merkuryev *Editors*

Simulation and Modeling Methodologies, Technologies and Applications

International Conference, SIMULTECH 2016, Lisbon, Portugal, July 29–31, 2016, Revised Selected Papers

Springer

Advances in Intelligent Systems and Computing

Volume 676

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

About this Series

The series "Advances in Intelligent Systems and Computing" contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within "Advances in Intelligent Systems and Computing" are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

Advisory Board

More information about this series at http://www.springer.com/series/11156

Mohammad S. Obaidat · Tuncer Ören
Yuri Merkuryev
Editors

Simulation and Modeling Methodologies, Technologies and Applications

International Conference, SIMULTECH 2016, Lisbon, Portugal, July 29–31, 2016, Revised Selected Papers

Springer

Editors
Mohammad S. Obaidat
King Abdullah II School of Information Technology
The University of Jordan
Amman
Jordan

Yuri Merkuryev
Department of Modelling and Simulation
Riga Technical University
Riga
Latvia

Tuncer Ören
School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa, ON
Canada

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-319-69831-1 ISBN 978-3-319-69832-8 (eBook)
https://doi.org/10.1007/978-3-319-69832-8

Library of Congress Control Number: 2017956736

Printed on acid-free paper

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

Preface

The present book includes extended and revised versions of a set of selected papers from the 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2016), held in Lisbon, Portugal, in the period July 29–31, 2016.

SIMULTECH 2016 received 76 paper submissions from 35 countries, of which 29% were included in this book. The papers were selected by the event chairs, and their selection is based on a number of criteria that include the reviews and suggested comments provided by the program committee members, the session chairs' assessments, and also the program chairs' global view of all papers included in the technical program. The authors of selected papers were then invited to submit a revised and extended version of their papers having at least 30% new material.

The purpose of the 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2016) was to bring together researchers, engineers, applied mathematicians, and practitioners interested in the advances and applications in the field of system simulation. Four simultaneous tracks were held, covering on one side domain independent methodologies and technologies and on the other side practical work developed in specific application areas. The specific topics listed under each of these tracks highlight the interest of this conference in aspects related to computing, including Conceptual Modeling, Agent-Based Modeling and Simulation, Interoperability, Ontologies, Knowledge-Based Decision Support, Petri Nets, Business Process Modeling and Simulation.

The papers selected to be included in this book contribute to the understanding of relevant trends of current research on Simulation and Modeling Methodologies, Technologies and Applications, including Discrete-Event Simulation, Image Simulation, Cluster Simulation, and Agent-based Simulation.

We would like to thank all the authors for their contributions and also to thank the reviewers who have helped ensuring the quality of this publication.

April 2017

Mohammad S. Obaidat
Tuncer Ören
Yuri Merkuryev

Organization

Conference Chair

Mohammad S. Obaidat	The University of Jordan, Jordan

Program Chair

Yuri Merkuryev	Riga Technical University, Latvia

Program Committee

Magdiel Ablan	Universidad de Los Andes, Venezuela
Nael Abu-Ghazaleh	University of California, Riverside, USA
Carole Adam	LIG - UJF, France
Lyuba Alboul	Sheffield Hallam University, UK
Mikulas Alexik	University of Zilina, Slovak Republic
Manuel Alfonseca	Universidad Autonoma de Madrid, Spain
Achraf Ammar	International Institute of Technologie IIT, Tunisia
Jan Awrejcewicz	The Technical University of Lódz (TUL), Poland
Gianfranco Balbo	University of Torino, Italy
Bartosz Balis	AGH University of Science and Technology, Poland
Simonetta Balsamo	University of Venezia Ca' Foscari, Italy
Isaac Barjis	City University of New York, USA
Jordi Mongay Batalla	National Institute of Telecommunications, Poland
Mohamed Bettaz	Philadelphia University, Jordan
Louis Birta	University of Ottawa, Canada
Wolfgang Borutzky	Bonn-Rhein-Sieg University of Applied Sciences, Germany

Christos Bouras	University of Patras and CTI&P Diophantus, Greece
Felix Breitenecker	Vienna University of Technology, Austria
António Carvalho Brito	INESC TEC, Faculdade de Engenharia, Universidade do Porto, Portugal
Hajo Broersma	University of Twente, the Netherlands
Christian Callegari	RaSS National Laboratory-CNIT, Italy
Jesus Carretero	Computer Architecture Group, University Carlos III of Madrid, Spain
Rodrigo Castro	University of Buenos Aires, Argentina
Krzysztof Cetnarowicz	AGH University of Science and Technology, Poland
Srinivas Chakravarthy	Kettering University, USA
Franco Cicirelli	Università della Calabria, Italy
Tanja Clees	Fraunhofer Institute for Algorithms and Scientific Computing (SCAI), Germany
Andrea D'Ambrogio	Università di Roma "Tor Vergata," Italy
Anatoli Djanatliev	Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
Atakan Dogan	Anadolu University, Turkey
Zhihui Du	Tsinghua University, China
Julie Dugdale	Laboratoire d'Informatique de Grenoble, France
Sabeur Elkosantini	King Saud University, Saudi Arabia
Zuhal Erden	ATILIM University, Turkey
Denis Filatov	Institute of Physics of the Earth, Russian Academy of Sciences, Russian Federation
Paul Fishwick	University of Texas at Dallas, USA
Jason Friedman	Tel-Aviv University, Israel
Richard Fujimoto	Georgia Institute of Technology, USA
Marco Furini	Università di Modena e Reggio Emilia, Italy
José Manuel Galán	Universidad de Burgos, Spain
Petia Georgieva	University of Aveiro, Portugal
Charlotte Gerritsen	Vrije Universiteit Amsterdam, the Netherlands
Luis Gomes	UNL/UNINOVA, Portugal
John (Yannis) Goulermas	The University of Liverpool, UK
Alexandra Grancharova	University of Chemical Technology and Metallurgy, Bulgaria
Mykola Gusti	International Institute for Applied Systems Analysis, Austria
Sigurdur Hafstein	University of Iceland, Iceland
Cathal Heavey	University of Limerick, Ireland
Monika Heiner	Brandenburg University of Technology Cottbus, Germany
Brian Hollocks	Bournemouth University, UK

Tsan-Sheng Hsu	Institute of Information Science, Academia Sinica, Taiwan
Xiaolin Hu	Georgia State University, USA
Eric Innocenti	IUT DE CORSE - University of Corsica, France
Nobuaki Ishii	Kanagawa University, Japan
Mhamed Itmi	INSA, Rouen, France
Mura Ivan	University EAN, Colombia
Yumi Iwashita	Kyushu University, Japan
Shafagh Jafer	Embry-Riddle University, USA
András Jávor	Budapest University of Technology and Economics, Hungary
Catholijn Jonker	Delft University of Technology, the Netherlands
Anniken Karlsen	NTNU in Ålesund, Norway
Peter Kemper	College of William and Mary, USA
Juš Kocijan	Jozef Stefan Institute, Slovenia
Petia Koprinkova-Hristova	Institute of Information and Communication Technologies, Bulgaria
Sunil Kothari	HP Labs, USA
Vladik Kreinovich	University of Texas at El Paso, USA
Claudia Krull	Otto-von-Guericke University, Germany
Pierre L'Ecuyer	Universite de Montreal, Canada
Mike Lees	University of Amsterdam, the Netherlands
Alberto Leva	Politecnico di Milano, Italy
Bin Li	University of Science and Technology of China, China
Fengyuan Li	Freescale Semiconductors, USA
Richard Lipka	University of West Bohemia, Czech Republic
Margaret Loper	Georgia Tech Research Institute, USA
Antonio Mendes Lopes	University of Porto, Portugal
Maria Celia Santos Lopes	COPPE-UFRJ, Brazil
Ulf Lotzmann	University of Koblenz, Germany
Johannes Lüthi	FH Kufstein Tirol, Austria
James F. Lutsko	Université Libre de Bruxelles, Belgium
José Tenreiro Machado	Institute of Engineering, Polytechnic of Porto, Portugal
Emilio Jiménez Macías	Universidad de la Rioja, Spain
Maciej Malawski	AGH University of Science and Technology, Poland
Andrea Marin	University of Venice, Italy
Carla Martin-Villalba	UNED, Spain
Radek Matušu	Tomas Bata University in Zlin, Czech Republic
Roger McHaney	Kansas State University, USA
Nuno Melão	Instituto Politécnico de Viseu, Escola Superior de Tecnologia e Gestão de Viseu, Portugal
Yuri Merkuryev	Riga Technical University, Latvia

Adel Mhamdi	RWTH Aachen University, Germany
Bozena Mielczarek	Wroclaw University of Technology, Poland
Federico Milani	CHEM.CO Consultant, Italy
Michael Möhring	University of Koblenz, Germany
Roberto Montemanni	IDSIA, Dalle Molle Institute for Artificial Intelligence (USI-SUPSI), Switzerland
Jairo R. Montoya-Torres	Universidad de los Andes, Colombia
Tingting Mu	University of Manchester, UK
Bertie Müller	University of South Wales, UK
Nazmun Nahar	University of Jyvaskyla, University of Tartu, Finland
Angela Nebot	Universitat Politècnica de Catalunya, Spain
Guyh Dituba Ngoma	Université du Québec en Abitibi-Témiscamingue, Canada
Lialia Nikitina	Fraunhofer Institute for Algorithms and Scientific Computing (SCAI), Germany
Michael J. North	Argonne National Laboratory, USA
Paulo Novais	Universidade do Minho, Portugal
James J. Nutaro	Oak Ridge National Laboratory, USA
Sorin Olaru	CentraleSupélec, France
Feng Pan	Liaoning Normal University, China
Ioannis Paraskevopoulos	Anglia Ruskin University, UK
George Pavlidis	"Athena" Research Centre, Greece
Krzysztof Pawlikowski	University of Canterbury, New Zealand
Alessandro Pellegrini	Sapienza, University of Rome, Italy
Alexandre Petrenko	Centre de Recherche Informatique de Montreal, Canada
Régis Plateaux	SUPMECA, France
Katalin Popovici	The MathWorks, Inc., USA
Tomas Potuzak	University of West Bohemia, Czech Republic
Francesco Quaglia	Sapienza Università di Roma, Italy
Martin Quinson	Université de Lorraine, France
Jacinto A. Dávila Quintero	Universidad de Los Andes, Venezuela
Manuel Resinas	Universidad de Sevilla, Spain
Jerzy Respondek	Silesian University of Technology, Poland
M.R. Riazi	Kuwait University, Kuwait
José Risco-Martín	Universidad Complutense de Madrid, Spain
Oliver Rose	Universität der Bundeswehr München (University of the Federal Armed Forces Munich), Germany
Ella E. Roubtsova	Open University of the Netherlands, the Netherlands
Willem Hermanus le Roux	CSIR, South Africa
Katarzyna Rycerz	Institute of Computer Science, AGH, Krakow, Poland, Poland

Cristina Montañola Sales	Universitat Politècnica de Catalunya, Spain
Janos Sallai	Vanderbilt University, USA
Paulo Salvador	Instituto de Telecomunicações, DETI, University of Aveiro, Portugal
Jean-François Santucci	SPE UMR CNRS 6134, University of Corsica, France
Florence Sèdes	IRIT, France
Peer-Olaf Siebers	University of Nottingham, UK
Flavio S. Correa Da Silva	University of Sao Paulo, Brazil
Jaroslav Sklenar	University of Malta, Malta
Andrzej Sluzek	Khalifa University, United Arab Emirates
John A. Sokolowski	Old Dominion University, USA
Xiao Song	Beihang University, China
James C. Spall	The Johns Hopkins University, USA
Giovanni Stea	University of Pisa, Italy
Bernhard Steffen	TU Dortmund University, Germany
Mu-Chun Su	National Central University, Taiwan
Nary Subramanian	University of Texas at Tyler, USA
Peter Summons	University of Newcastle, Australia
Antuela A. Tako	Loughborough University, UK
Halina Tarasiuk	Warsaw University of Technology, Poland
Pietro Terna	Università di Torino, Italy
Constantinos Theodoropoulos	University of Manchester, UK
Mamadou K. Traoré	Blaise Pascal University, France
Klaus G. Troitzsch	University of Koblenz-Landau, Koblenz Campus, Germany
Zhiying Tu	Harbin Institute of Technology, China
Kay Tucci	Universidad de los Andes, Venezuela
Alfonso Urquia	Universidad Nacional de Educación a Distancia, Spain
Timo Vepsäläinen	Space Systems Finland, Finland
Maria Joao Viamonte	Instituto Superior de Engenharia do Porto, Portugal
Manuel Villen-Altamirano	Universidad de Malaga, Spain
Friederike Wall	Alpen-Adria-Universität Klagenfurt, Austria
Hao Wang	Norwegian University of Science and Technology, Alesund, Norway
Frank Werner	Otto-von-Guericke-Universität Magdeburg, Germany
Hannes Werthner	Technical University of Vienna, Austria
Philip A. Wilsey	Univ. of Cincinnati, USA
Kuan Yew Wong	Universiti Teknologi Malaysia, Malaysia
Li Xia	Tsinghua University, China
Yiping Yao	National University of Defense Technology, China

Gregory Zacharewicz	University of Bordeaux, France
František Zboril	Faculty of Information Technology, Czech Republic
Durk Jouke van der Zee	University of Groningen, the Netherlands
Houxiang Zhang	Norwegian University of Science and Technology, Norway
Qianchuan Zhao	Tsinghua University, China
Suiping Zhou	Middlesex University, UK
Armin Zimmermann	Technische Universität Ilmenau, Germany

Additional Reviewer

Jeremy Sproston	Università degli Studi di Torino, Italy

Invited Speakers

Francesco Casella	Politecnico di Milano, Italy
Catholijn Jonker	Delft University of Technology, the Netherlands
Yaman Barlas	Bogaziçi University, Turkey

Contents

Invited Paper

Credibility, Validity and Testing of Dynamic Simulation Models

Yaman Barlas(✉)

Industrial Engineering Department, SESDYN Lab, Boğaziçi University,
34342 Bebek, Istanbul, Turkey
ybarlas@boun.edu.tr
http://www.ie.boun.edu.tr/labs/sesdyn/

Abstract. Also called 'model validity' testing, model credibility evaluation has always been a controversial issue in any modeling methodology. We briefly discuss why this important notion is so controversial. To this end, we classify major types of models, particularly as they impact the notion of model credibility: i- Purely statistical forecasting (black box) models, and ii- Causal-descriptive policy (transparent) models. We then focus on what makes causal-descriptive model credibility and evaluation unique and quite difficult, compared to short-term forecasting models. One important result is that causal-descriptive model credibility consists of two different aspects: *structural* and *behavioral*. In most simulation modeling (particularly system dynamics policy simulation), establishing *structure credibility* must strictly precede *behavior credibility*; the latter has no value without the former. We thus discuss *Structural tests* and *output behavior tests* for dynamic simulation models separately. *Structure* tests can further be classified into *direct* and *indirect* structure tests. We place special emphasis on *indirect* structure tests. We also provide a quick overview of recent model testing software developed in SESDYN Laboratory at Boğaziçi University. Finally we discuss model credibility in broader context and some implementation issues.

1 Introduction

Model credibility and validity have long been recognized as one of the main issues in simulation and modeling in general [1, 3, 5, 9–11, 13–17, 19, 20, 21, 23]. A main difficulty comes from the fact that two different types of models necessitate two very different approaches to model credibility testing. Credibility of a causal-descriptive (theory-like, "transparent") model is critically different from that of a purely correlational (data-driven, "black-box") model [3, 5]. In purely correlational (black-box) modeling, since there is no claim of a causal, meaningful structure, the model is assessed valid if its output behavior matches real output data within some specified range of accuracy, without any questioning of the credibility of the relationships that constitute the model. For such models, model validity essentially means the validity of the output behavior. Models that are built primarily for short term forecasting purpose (such as time-series or regression models) belong to this category. On the other hand, causal-descriptive (transparent) models are hypotheses as to how real systems actually

M.S. Obaidat et al. (eds.), *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing 676,
https://doi.org/10.1007/978-3-319-69832-8_1

operate in creating the dynamics of interest. In this case, generating an "accurate" output behavior is not sufficient for model credibility; what is crucial is the validity of the *internal structure* of the model. The model *structure* is defined to be the totality of the relationships that exist in the model. A descriptive policy-oriented model must not only reproduce/predict the real behavior, but must also explain how the behavior is generated, and be able to suggest ways of improving the existing behavior. Most dynamic simulation models, particularly policy-design-oriented ones (such as system dynamics models) fall in this category.

In this article we focus on credibility of causal-descriptive simulation (transparent) models. Since these models are hypotheses about how systems actually operate in real life, testing of such models faces a fundamental philosophical challenge: is it possible to 'prove' conclusively the truth of any given scientific hypothesis? This fundamental and unresolved philosophy of science question is discussed in the context of model validation by Barlas and Carpenter [5]. To summarize very briefly, there are two opposing schools in approaching this question: The *logical positivist (empiricist)* school argues that by a proper method and enough data, it is philosophically possible to prove conclusively the truth of a scientific hypothesis. The implication of this radical philosophy for model validation is that a model is meaningless unless its validity is 'proven' by a standard method and enough data. There is no gray area, no 'degrees of validity' and any claim of validity must be supported by data. According to the opposing (relativist, conversationalist) philosophical school however, truth of a scientific hypothesis can never be positively and conclusively proven, no matter what method is used and how much data are processed. 'Truth' of scientific hypotheses do not have final and rigid 'yes or no' answers; hypotheses are assumed to be 'true' temporarily and a result of a holistic (scientific and social; objective and subjective) process, until better hypotheses are developed. The implication for model validity is that there are degrees of model validity established gradually by multiple inputs (both quantitative and qualitative data, both scientific and social), and a critical dimension of model credibility is its *usefulness* with respect to a *purpose*. In this paper we submit that this relativist-conversationalist philosophy is more appropriate in discussing credibility of simulation models.

As mentioned above, validation of a causal-descriptive simulation model consists of two main components: *structure* testing and *behavior* testing. *Structure* validation means establishing that the relationships used in the model are an adequate representation of the real relationships, with respect to the purpose of the study. *Behavior* validation consists of demonstrating that the behavior of the model is "close enough" to the observed real behavior. For transparent models, there is no point in testing the behavior validity, until the model demonstrates an acceptable level of structure validity. The model would be refuted if a relationship in the model conflicts with a known/established "real relationship", even if the output behavior of the model matches well the observed system behavior. For causal-descriptive models, validity ultimately means credibility of the internal structure of the model, and structure validation must precede behavior validation. (See [3] for more discussion).

In the following sections, we first discuss *structure* credibility, its different dimensions and some specific test methods. Next we discuss *output behavior*

credibility and some selected tests and tools. We conclude with some observations on implementation problems, avenues for improvements and further research.

2 Structure Credibility Testing

As defined above, *structure* is defined to be the totality of relations that exist in a system (model or real system). Testing the credibility of model structure, as explained above, is the essence of model credibility for causal-descriptive models, so it must precede output behavior validity testing. (See Fig. 1). Testing the credibility of model structure consists of two types of tests: 1- *direct* structure testing, 2- *indirect* structure testing [3]. *Direct* structure tests assess the credibility of the model structure, by direct comparison with knowledge about real system structure. This involves taking each relationship (mathematical equation or some form of relationship) individually and comparing it with available knowledge about real system. There is no simulation involved and these tests are highly qualitative in nature. *Indirect* structure tests, on the other hand assess the credibility of the structure indirectly, by applying some special behavior tests on model-generated behavior patterns. (See [7, 11]). Different than direct structure tests, these tests involve simulations, so they are relatively more quantitative in nature.

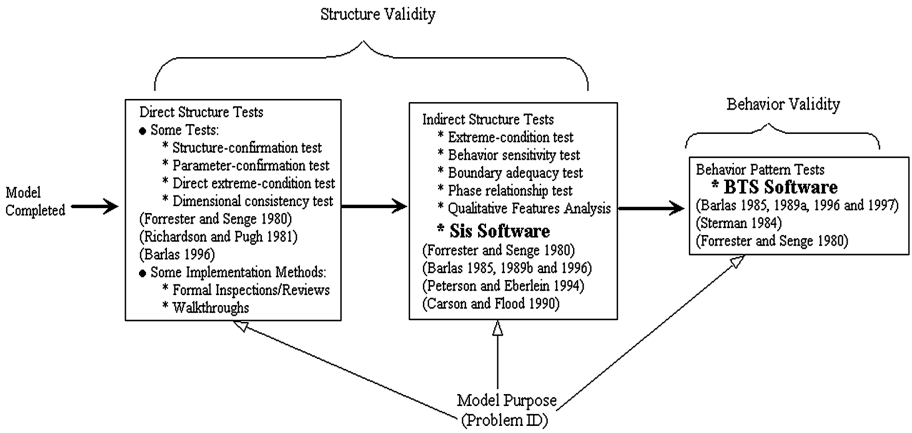


Fig. 1. The main components and logical order of model credibility testing.

2.1 Direct Structure Tests

Direct assessment of model structures is crucial in model credibility, but the difficulty is that such tests are highly qualitative and judgmental by nature. For instance, 'structure confirmation test' involves taking each equation and asking: does such a (similar) relationship exist in reality? [3, 11]. To establish structure validity, the answer to this question must be 'yes' for each equation in the model. But answering this question cannot be quantified, nor can it be automated. Different people may express different opinions in such tests. Hence, direct structure tests are critically important conceptually,

but weak in application. Another important direct structure test is 'parameter confirmation' test that requires that each parameter in each equation has a real life meaning that can be explained and defended -in other words meaningless, dummy parameters are not allowed. There are two direct structure tests that are particularly important, because they are more quantitative and objective in nature. The first one is 'unit consistency test' that requires that the units of variables and parameters in each equation must automatically (naturally) give the unit of the left hand side variable, without using any dummy parameters to make the units match [3, 11, 22]. This test is an absolute requirement that all equation-based models must pass. Also note that this test is strongly related to (must be applied together with) the parameter confirmation test above. The other important and relatively quantitative direct structure test is 'direct extreme condition test' that requires that each equation must yield 'logically defendable' extreme values, when the input variables and/or parameters are set at extremes [3, 11, 22]. This is in a sense a 'stress' test applied to each equation; if the equation has a logical weakness (hidden in normal operating range of inputs), the weakness may be revealed under extreme input conditions. Some other direct structure tests are shown in Fig. 1, in the first box. By their nature, the principles of direct structure credibility and related tests are much more useful when they are applied as 'model equation writing principles' during model construction (rather than as 'validity tests' after the model has been completed. This point will be discussed below, under Building High-quality Models versus Testing Structure Validity.

2.2 Indirect Structure Tests

Indirect structure tests assess the validity of model structure indirectly, by applying selected behavior tests on model-generated behavior patterns. (See [7, 11]). These tests involve simulation, and can be applied to the entire model, as well as to isolated model sub-structures. For example, extreme-condition simulation test involves assigning extreme values to selected parameters and comparing the model-generated behaviors to the "anticipated" (or observed) behaviors of the real system under the same (or similar) extreme condition. (See [7] for illustrations). Indirect structure tests can be interpreted as strong behavior tests that can provide information on potential structural flaws. Their main advantage over direct structure tests is that they are much more suitable to formalize and quantify, since they involve simulations and output assessments. Other early examples of indirect structure tests include boundary adequacy test, phase relationship test [11], "Qualitative Features Analysis" by Carson & Flood [9] and the "Reality Check" feature of VENSIM simulation software [18]. In this article, we present a general method and software that we developed for indirect structure testing.

'Indirect structure testing software' (ISTS) is a computerized algorithm that seeks to automate indirect structure credibility testing [12]. In indirect structure tests, the modeler makes a claim of the form: "if the system operated under some condition C, the behavior B should result". The model is then simulated under condition C and is said to "pass" this test, if the resulting model behavior is similar to the expected behavior B. In the automated structure-oriented behavior testing environment, the modeler hypothesizes a dynamic behavior by choosing a dynamic pattern from a template of all basic patterns (summarized in Fig. 2). The computerized algorithm then

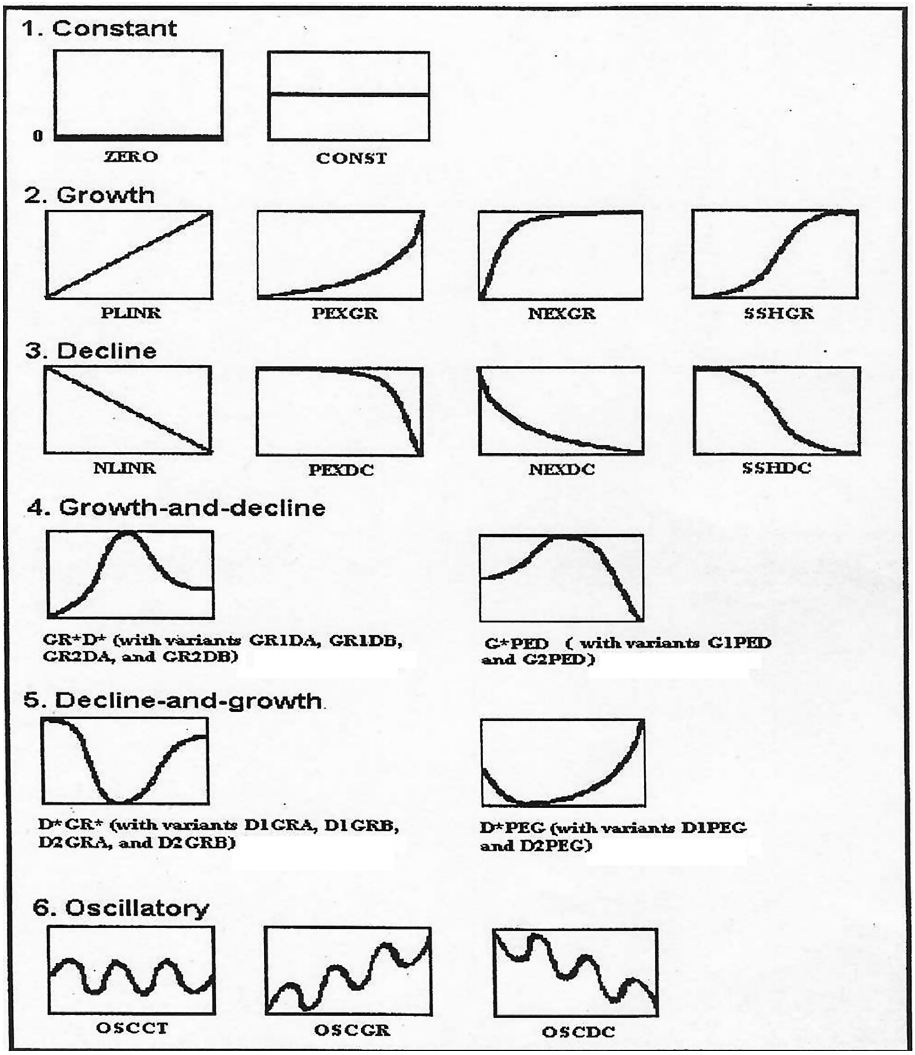


Fig. 2. Template of basic dynamic patterns.

takes the dynamic behavior generated by the model, "recognizes" it and tests if it belongs to the pattern class hypothesized by the modeler. ISTS is thus essentially a 'dynamic pattern recognition' algorithm. The algorithm used is a pattern recognizer/classifier based on Hidden Markov Models (HMM). The statistical pattern recognition is achieved by a classification of feature vectors extracted from the data. Hidden Markov Models do not represent the whole data with a single feature vector. In HMM-based pattern recognition, one dimensional data is divided into segments and a sequence of feature vectors is extracted. (An alternative approach to dynamic pattern recognition problem is presented in [25, 26]). In the dynamic behavior recognition

problem, a dynamic behavior can be denoted by a sequence *y(k), k = 1, 2, ..., K*, where *K* is the number of data points. As depicted in Fig. 3, such a signal would be a somewhat distorted (or "noisy") version of one of the patterns given in the template of Fig. 2. The next step is to extract features from each data segment. Basic dynamic patterns are characterized by successive time segments of growth or decline and their trends (as growing or declining rates). Therefore, it is reasonable to form the feature vector using the slope and 2nd derivative ("curvature") information of the data in each segment. The features can be obtained by fitting polynomials to each segment data. The slope of the first-order polynomial provides trend information, which is growth, decline or constant. The second-order polynomial can be used to obtain the second derivative, which will yield the curvature information. In addition to slope and curvature, the level of the selected variable also provides useful information. Thus, the segment mean level becomes the third element of the feature vector. In summary, each feature vector is M = 3 dimensional and given by three components; slope, curvature, and mean. By comparing the feature vectors computed from a given output behavior (as in Fig. 3) to the feature vectors that characterize the predefined pattern classes (as seen in Fig. 2), ISTS computes the likelihoods of the given dynamic behavior to belong to each predefined class. Finally, the given behavior is 'recognized' and classified into the pattern class that maximizes the likelihood. The general flowchart of ISTS algorithm is given in Fig. 4.

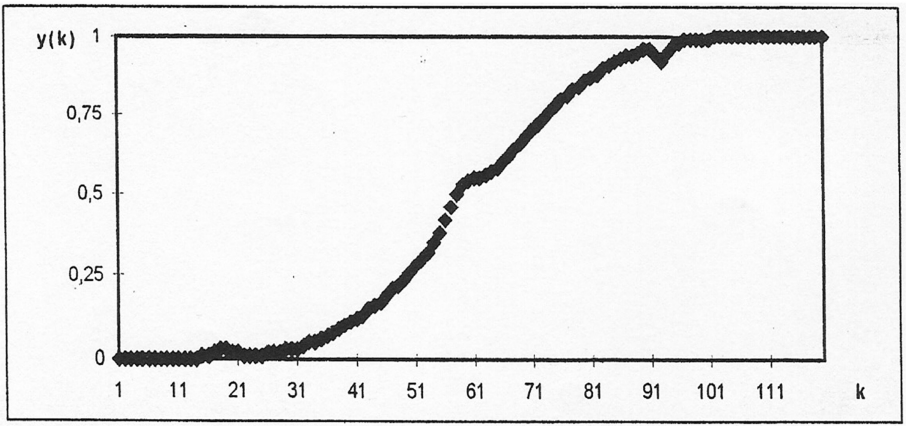


Fig. 3. A dynamic output behavior example.

Based on ISTS algorithm, a user-friendly indirect structure validity testing and calibration software (SiS) has been developed [2]. The purpose is to bridge ISTS algorithm and the existing VENSIM simulation software and automate the validity testing and parameter calibration of dynamic simulation models. The software is written in JAVA programming language. SiS software consists of validity testing and automatic parameter calibration functions. Figure 5 illustrates the general structure of the validity testing function of SiS.

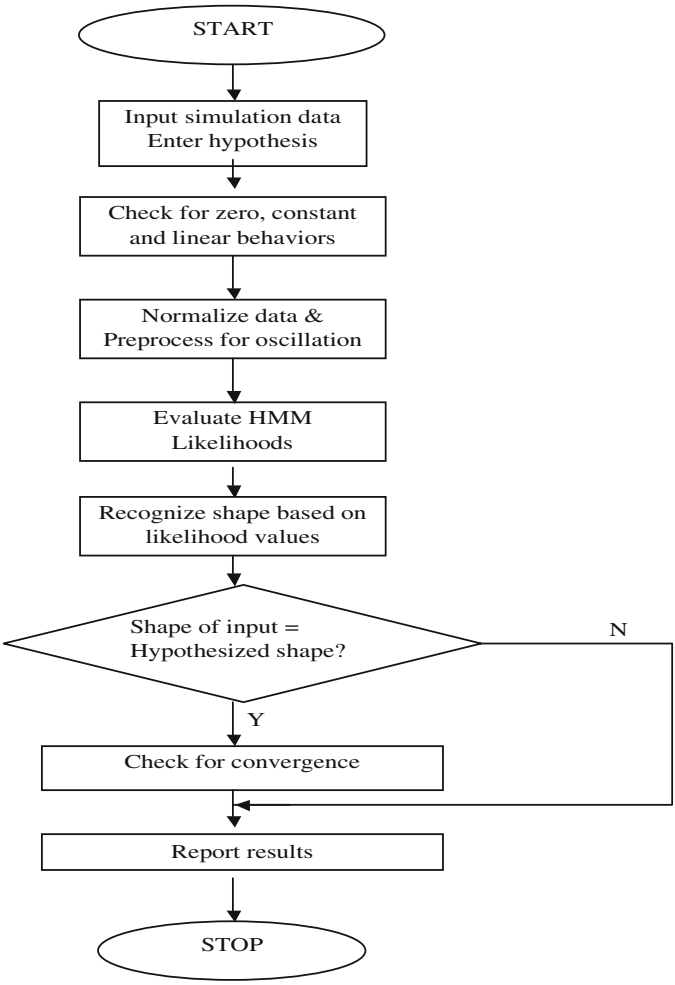


Fig. 4. The flowchart of ISTS pattern recognition-classification algorithm.

The user first hypothesizes the output behavior pattern expected from the model in certain conditions (like some extreme condition). The hypothesized output pattern is selected from a template of existing basic patterns, like the ones shown in Fig. 2. "Integrator" part of SiS loads the selected model to VENSIM and issues the command to start simulation. Simulation output behavior generated by VENSIM model is read back by "integrator". "Main" part takes the simulation output pattern (like the one in Fig. 3) and executes ISTS Algorithm to perform the validity test. If the output behavior is classified in the hypothesized pattern class, then the test is passed, else it is failed. For instance, the output behavior shown n Fig. 3 would be recognized and classified by SiS as SSHGR pattern (shown in Fig. 2). SiS also has an automated model calibration and policy analysis function that determines those parameter values that yield the output dynamics that fit best to the desired pattern class [2, 25].

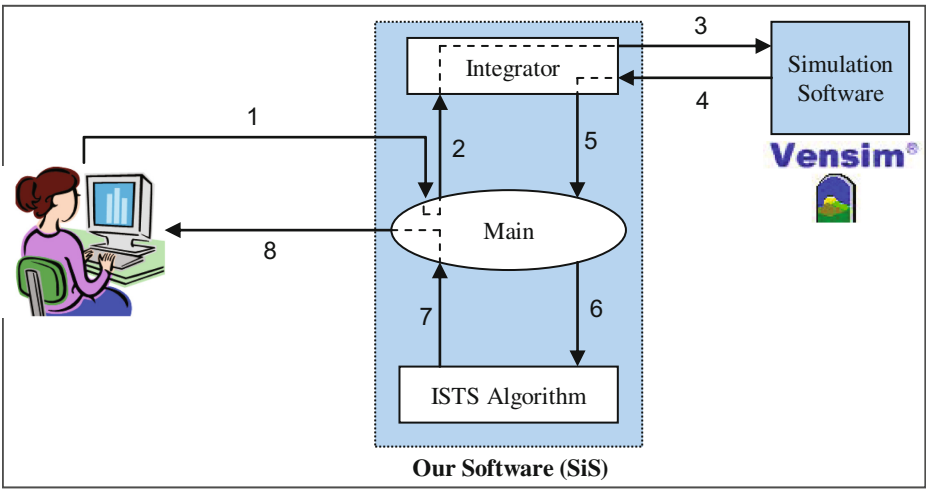


Fig. 5. General structure of indirect structure testing function of SiS.

2.3 Building High-Quality Models Versus Testing Model Validity

The above sections summarized direct and indirect model structure tests and some related software. Another important dimension of structure credibility is that it is most critical to use of the principles of (direct) structure credibility during model construction. This way, to adopt a well-known quality control principle, model credibility will be 'built-in' rather than 'inspected in' later by model testing. The main principles of high model credibility that must be followed during model construction are:

- Problem statement and model purpose must be clear and specific
- Time unit, model resolution, aggregation and time horizon must all be properly and consistently selected in the light of model purpose
- All variables and parameters must have real life meanings
- All variables and parameters must have meaningful units
- All equations and logical expressions must have explainable meanings
- All equations must have natural unit consistency, without using any dummy parameters to make the units match
- Equations must yield logical extreme results under extreme input conditions
- Established formulations and structures in related literature must be used
- Correctness of the simulation program must be verified: Does the simulation model correctly represent my conceptual model? Does it have any unintended, careless errors, including simple typing errors?
- If the model is time-continuous, is the time step (dt) small enough to rule out any spurious dynamics caused by numeric errors?
- It is best to start with a small model and embellish gradually, one structure at a time, by using partial structure tests in each step
- Good model documentation is crucial for others to assess the quality of the model (establishing model credibility also means convincing other people).

The above list highlights the most important structure validity principles particularly for equation-based simulation models (like system dynamics, as discussed in [22]). The list can be properly modified for other types of simulation models. For any type of causal-descriptive modeling, if these principles are utilized during model building, a high quality (hence highly credible) model can be constructed to start with, which is much more effective than carrying out the structure credibility tests at the end, after the model is completed.

3 Output Behavior Credibility Testing

After enough confidence is established in the credibility of model structure, output behavior validity tests can be carried out (last box, Fig. 1). The purpose of behavior tests is to demonstrate that the dynamics of the model are "close enough" to the observed real dynamics. However, the match between the model dynamics and real data can be measured in two very different ways: i- point-by-point match, ii- dynamic 'pattern' match. In most regression models and short-term forecasting studies, validity of the model is judged by how well it matches the real data on a point-by-point basis. But the situation is very different for long-term policy-oriented simulation studies (such as system dynamics). In such studies, it is nether possible, nor meaningful to expect a good point-by-point match between the model dynamics and real data. These causal-descriptive models start with a set of initial conditions and generate the long-term dynamics of the system endogenously, not by 'curve fitting'. The purpose is not to provide short-term forecasts by optimum curve fitting, but to project long-term dynamic consequences of adopted policies [6, 11, 22]. In short, the purpose is to provide long-term pattern predictions, so the validity must be measured by how well the real dynamic patterns (periods, frequencies, trends, phase lags, amplitudes…) are reproduced/predicted by the model [4, 6, 11]. Thus, suitable statistical tools and methods focusing on dynamic pattern components are needed. One such method and software is 'Behavior Testing Software II' (BTS II) developed at Boğaziçi University, SESDYN Laboratory [8]. To start with, two very different types of patterns are treated separately by BTS II: *steady-state* patterns can be processed by suitable statistical tools, whereas *transient* patterns can not be measured or analyzed statistically, since they consist of non-stationary, non-repetitive features. Periods and amplitudes of oscillations, mean levels, long-term trend slopes are examples of stationary measures for which BTS II provides statistical tools. Transient dynamics on the other hand can be characterized by features like maxima, minima, inflection points that can be measured graphically (see Fig. 6). After this initial separation of the model dynamics, BTS II can be used to measure the relevant patterns of model dynamics and real data and then compare them to assess how close the pattern components are (by comparing trends, periods, amplitudes, autocorrelation functions, etc.). Transient pattern features can be measured and compared by the graphical tools provided by BTS II. If the test results are not satisfactory, the model parameter values and/or certain model structures may have to be revised by the analyst.

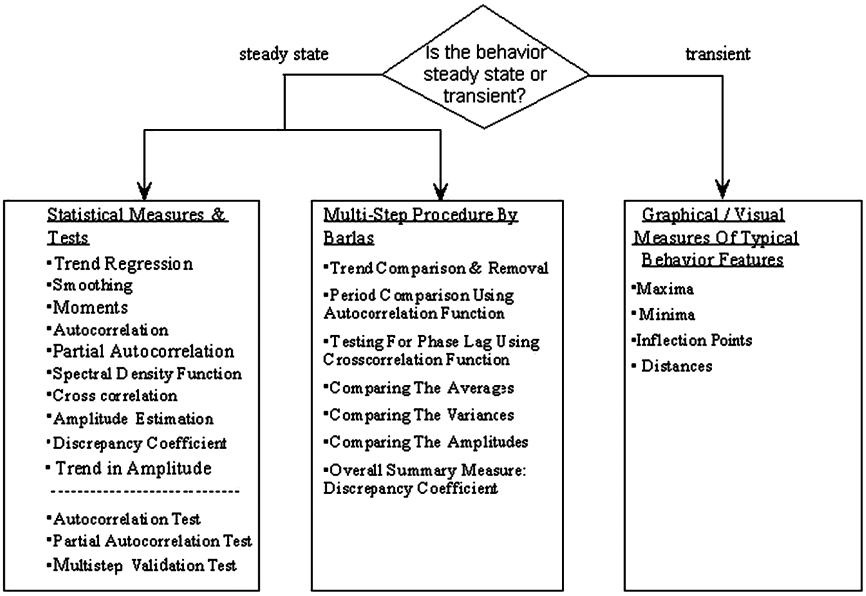


Fig. 6. Overview of behavior credibility testing software (BTS II).

4 Concluding Observations

For meaningful and productive model credibility testing, the modeler must be proficient in two dimensions: in conceptual and philosophical foundations, and in accessing and using relevant tools and software. In this article we first discussed the conceptual and philosophical foundations. We then provided an overview of available tools and software that can be used in different phases of testing model credibility, namely structural and behavioral. For wide spread and standardized use of tools and software, there is need for more user-friendly model testing software that can directly and easily communicate with the existing simulation modeling software. A recent step in this direction is BATS (Behavior Analysis and Testing Software) developed at Boğaziçi University, SESDYN laboratory [24]. BATS integrates features of two tools and software reviewed in this article: SiS for indirect structure testing and BTS II for output behavior testing (see Fig. 8), has a user-friendly interface and can communicate directly with Vensim software (Fig. 7). But the final version of the software is not released yet, because there are still some communication problems with Vensim software and links with other simulation software are not established yet. Certainly more research is needed to develop new tools and software for model credibility testing.

In another dimension, as emphasized in the article, it is most critical to use of the principles of structure credibility during model construction (rather than testing the model after it is completed). To rephrase a well-known quality control principle, model credibility should be ‘built-in’ rather than ‘inspected in’ later by model testing. In this sense, ‘model quality’ is actually a much better term than ‘validity’ or ‘credibility’ in

evaluating simulation models. It is possible to argue that ideally the term *quality* should permanently replace the terms validity and credibility in simulation modeling.

Finally, one must keep in mind that model credibility cannot be established by purely formal tools, algorithms and software. Model credibility establishment is a prolonged, distributed process that starts with model construction and does not end till the implementation of model recommendations. It is a gradual, multi-dimensional process that involves not only the modeler/analyst, but also various stakeholders, which means that building confidence in model is unavoidably a social process. Credibility of policy recommendations and successful implementation is a very important and different problem in itself, not addressed in this article [11]. There are methods like group modeling and interactive simulation gaming to engage stakeholders to address the issues of successful policy design and real life implementation.

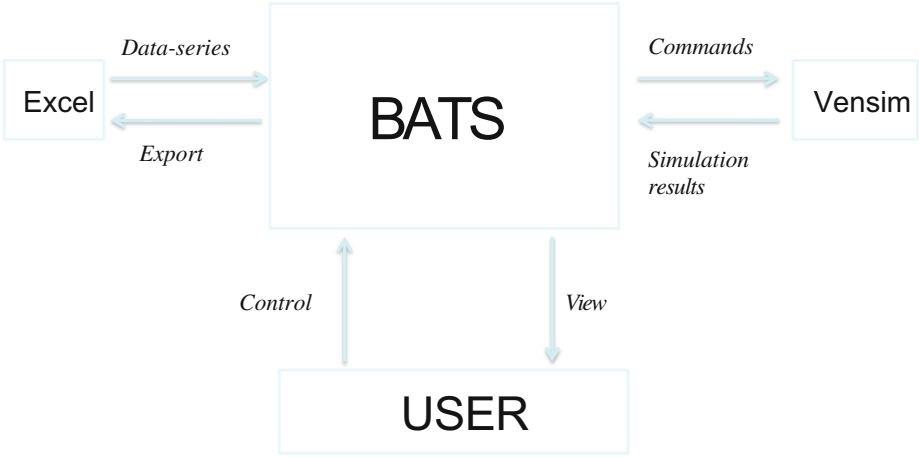


Fig. 7. Overview of BATS software.

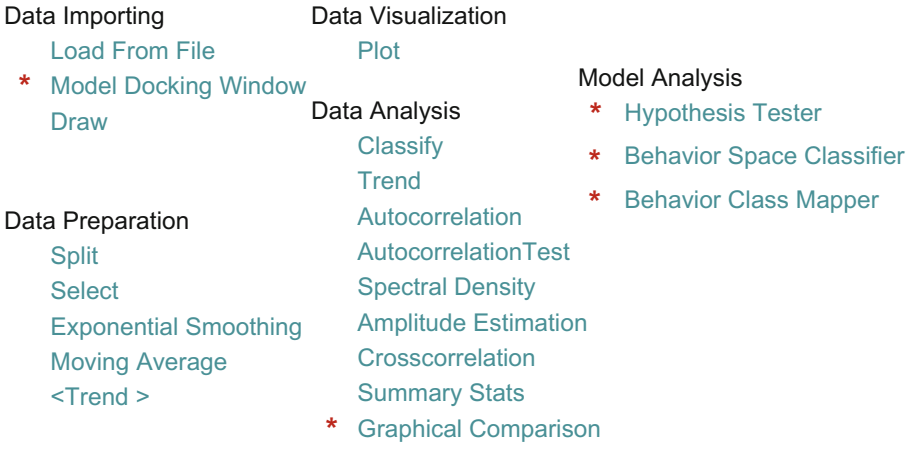


Fig. 8. Main features and functions of BATS software.

References

1. Balci, O.: Validation, verification and testing techniques throughout the life cycle of a simulation study. Ann. OR **53**, 121 (1994)
2. Barlas, Y., Boğ, S.: Automated dynamic pattern testing, parameter calibration and policy improvement. In: Proceedings of International System Dynamics Conference (CD), Albany, NY, USA (2005)
3. Barlas, Y.: Formal aspects of model validity and validation in system dynamics. Syst. Dyn. Rev. **12**(3), 183–210 (1996)
4. Barlas, Y.: An autocorrelation function test for output validation. Simulation **55**(1), 7–16 (1990)
5. Barlas, Y., Carpenter, S.: Philosophical roots of model validation: two paradigms. Syst. Dyn. Rev. **6**(2), 148–166 (1990)
6. Barlas, Y.: Multiple tests for validation of system dynamics type of simulation models. Eur. J. Oper. Res. **42**(1), 59–87 (1989)
7. Barlas, Y.: Tests of Model Behavior That Can Detect Structural Flaws: Demonstration With Simulation Experiments. In: Milling, P.M., Zahn, E.O.K. (eds.) Computer-Based Management of Complex Systems: International System Dynamics Conference. Springer, Heidelberg (1989)
8. Barlas, Y., Topaloğlu, H., Yılankaya, S.: A Behavior Validity Testing Software (BTS). In: Proceedings of the 1997 International System Dynamics Conference, Istanbul (1997)
9. Carson, E.R., Flood, R.L.: Model validation: philosophy, methodology and examples. Trans. Inst. MC **12**(4), 178–185 (1990)
10. Eur. J. Oper. Res. Special Issue on Model Validation **66**(2) (1993)
11. Forrester, J.W., Senge, P.M.: Tests for building confidence in system dynamics models. In: Legasto, A.A., Forrester, J.W., Lyneis, J.M. (eds.) System Dynamics. North-Holland, Amsterdam (1980)
12. Kanar, K., Barlas, Y.: A Dynamic pattern-oriented test for model validation. In: Proceedings of 4th Systems Science European Congress, Valencia, Spain, September 1999, pp. 269–286 (1999)
13. Kleijnen, J.P.C.: Verification and validation of simulation models. Eur. J. Oper. Res. **82**, 145–162 (1995)
14. Naylor, T.H., Finger, J.M.: Verification of computer simulation models. Manag. Sci. **14**, 92–101 (1968)
15. Oral, M., Kettani, O.: The facets of the modeling and validation process in operations research. Eur. J. Oper. Res. **66**(2), 216–234 (1993)
16. Ören, T.I., Yilmaz, L.: 'Philosophical Aspects of Modeling and Simulation' in Ontology, Epistemology, and Teleology for Modeling and Simulation. Springer, Heidelberg (2013)
17. Ören, T.I.: Concepts and criteria to assess acceptability of simulation studies: a frame of reference. Commun. ACM **24**(4), 180–189 (1981)
18. Peterson, D.W., Eberlein, R.L.: Reality check: a bridge between systems thinking and system dynamics. Syst. Dyn. Rev. **10**(2–3), 159–174 (1994)
19. Saysel, A.K., Barlas, Y.: Model simplification and validation with indirect structure validity tests. Syst. Dyn. Rev. **22**(3), 241–262 (2006)
20. Schruben, L.W.: Establishing the credibility of simulations. Simulation **34**(3), 101–105 (1980)
21. Sheng, G., Elzas, M.S., Ören, T.I., Cronhjort, B.T.: Model validation: a systemic and systematic approach. Reliab. Eng. Syst. Safety **42**(2), 247–259 (1993)

22. Sterman, J.D.: Business Dynamics: Systems Thinking and Modeling for a Complex World, p. 982. McGraw-Hill, New York (2000)
23. Sterman, J.D.: Testing behavioral simulation models by direct experiment. Manag. Sci. **33** (12), 1572–1592 (1987)
24. Sücüllü, C., Yücel, G.: Behavior analysis and testing software (BATS). In: Proceedings of the 32nd International Conference of the System Dynamics Society, Delft, Netherlands (2014)
25. Yücel, G., Barlas, Y.: Pattern recognition for model testing, calibration, and behavior analysis. In: Analytical Methods for Dynamic Modelers, pp. 173–206. MIT Press, Massachusetts (2015)
26. Yücel, G., Barlas, Y.: Automated parameter specification in dynamic feedback models based on behavior pattern features. Syst. Dyn. Rev. **27**(2), 195–215 (2011)

Papers

Model-Based Development of a Multi-algorithm Harvest Planning System

Luis Diogo Couto[1], Peter W.V. Tran-Jørgensen[1(✉)], and Gareth T.C. Edwards[2]

[1] Department of Engineering, Aarhus University, Aarhus, Denmark
pvj@eng.au.dk
[2] Agro Inteligence ApS, Aarhus, Denmark

Abstract. Planning systems for harvest operations need to employ complex algorithms to calculate various aspects of the harvest plan such as the order in which to harvest field rows or when and where to unload harvesters. In traditional modelling and simulation approaches, it is not easy to vary the algorithm as a simulation parameter. This either limits the solution space for a system or it forces significant duplication to set up various models with the necessary algorithms. In this paper, we present the Model-Based Development of a planning system that leverages the strategy pattern to enable efficient variation of the optimisation algorithms at various stages of the planning process. We illustrate the system by applying it to a real field and discuss issues such as coping with large fields and how to carry out a real harvest operation according to the plan.

1 Introduction

There are various steps to calculating optimised solutions for harvest operations. These steps include partitioning of the field and calculating optimised coverage plans for harvesters and route plans for other vehicles. One approach to the problem involves the use of various optimisation algorithms that produce coverage plans for the harvesters [1,2]. However, planning of harvester routes is just one part of the harvest operation planning. Path planning for grain wagons (or similar) that service the harvesters must often also be developed. Algorithms exist for optimising service plans [3] but they are independent from those of harvesters. This independence makes it difficult to explore in detail how the various types of algorithms interact and combine to produce a complete solution for the harvest operation.

As an example, little research has previously been conducted into how harvesting and loading algorithms can affect operational execution times of harvest operations. Examples of planning tools for operations often employ a single algorithm; such as in-field unloading [4] or single point unloading [5]. Farmers will generally choose a plan with which they are familiar without considering alternatives.

M.S. Obaidat et al. (eds.), *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing 676,
https://doi.org/10.1007/978-3-319-69832-8_2

In this paper, we seek to explore how different optimisation algorithms can be combined. We will explore this using a formal[1] model in combination with the strategy pattern from software engineering. The strategy pattern is used in the model to encode different optimisation algorithms. A novel aspect here is that the strategies representing the different kinds of algorithms (harvest routing and grain wagon path planning) co-exist and collaborate to produce the final solution.

From an operational research perspective, the harvest operation is an example of an output material flow (OMF) operation where material is removed from the field and transported to another location [6]. The machinery utilised within the OMF operation can be divided into two groups; Primary Units (PUs) which perform the main task i.e. harvesting the crop, and Service Units (SUs) which service the PUs by receiving harvested material and transporting it away. The capacity of the PU is many times smaller than the expected yield of the field, and therefore a PU unloads either to a nearby SU or directly to an out of field storage point.

The planning of the tasks of the PUs and SUs are often considered separately [7], with coverage plans being developed for PUs [1,2] and path plans being developed for grain wagons [3]. However, the tasks are spatially and temporally dependant on one another, so in order for efficient plans to be produced the plans must be developed concurrently [8].

To assist with the planning of in-field operations, fields can be decomposed into a number of tracks or rows. Many methods have been proposed for the decomposition of fields [4,9–11]. Fields are typically divided into headlands which encircle the field and can be used for turning, and working rows which transect the main area of the field. By confining all field traffic to drive along these predefined rows, the trafficked area of the field can be limited which has been shown to produce benefits on increased yield and better soil structure [12].

In the above mentioned approaches, the planning for the various kinds of vehicles is performed independently, as is the decomposition of the field. In our work, we consider all vehicles simultaneously when planning, although field decomposition is still done separately.

A different approach to optimisation was carried out in a EU project called DESTECS. In this project design space exploration is performed by sweeping parameters of models of cyber-physical systems [13]. Among other things, the DESTECS project proposes methodological guidelines for modelling fault-tolerant cyber-physical systems, which also involve the use of the strategy pattern to model faulty behaviour as well guarding against it [14]. This is similar to the presented approach, in that the strategy pattern is used in the DESTECS project to explore different behaviours of a system. However, while the DESTECS project used the strategy pattern to make a system more fault-tolerant, in this work the strategy pattern is used to help find optimised solutions to use in a harvest operation.

[1] *Formal* in this context means that the model is developed in a notation that is given semantics in a formal logic.

The strategy pattern is a design pattern [15] with two key features. First, the strategy pattern allows selection of different algorithms to be done at execution time and; secondly, it defines a family of interchangeable algorithms. Essentially, this allows the same functionality to be executed in different ways. Broadly speaking, the strategy pattern consists of a contract that defines the functions of a strategy in terms of their inputs and outputs including the properties that these functions may have. Given this contract, a specific strategy must provide an implementation of the functions that obeys the input and output properties of the contract, but which is free to use whatever algorithms are desired.

The remainder of this paper is structured as follows: in Sect. 2 we present the architecture of the formal model of the harvest operation based on the strategy pattern. The technologies that have been used to implement the model are described in Sect. 3. Next, the execution of the model is demonstrated in Sect. 4. Following that, in Sect. 5, we report the results of applying the model to a case study of a real field. The results are then discussed in Sect. 6. Afterwards, in Sect. 7, we describe how analysis of data reported by harvesters and grain wagons can be used to continuously optimise a plan over the course of a harvest. Finally, we conclude the paper in Sect. 8.

2 Model Architecture

2.1 Model Overview

The model was developed according to the structure shown in Fig. 1. The Execution Engine is responsible for coordinating the simulation and is connected to both the State and the three Strategy classes. The State contains the physical entities involved in the harvest operation. The harvesters are the PUs of the operation. Coverage plans and coordinated service points are developed for the harvesters by the employed strategies. The grain wagons are the SUs of the harvest operation and are used to convey material from the harvesters to the out-of-field storage. The service points coordinate when and where the grain wagons must meet the harvesters in order for material to be passed between the two.

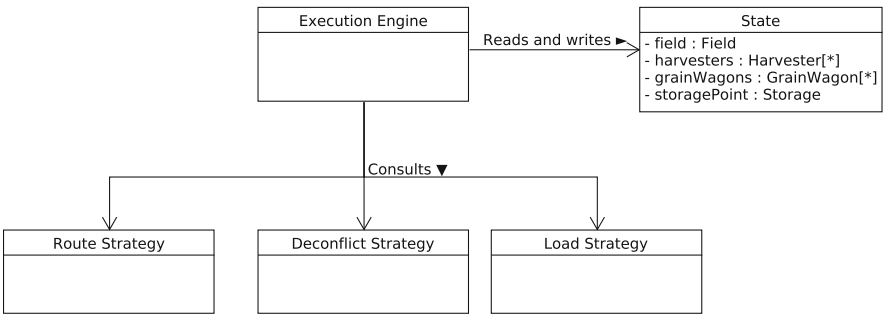


Fig. 1. Model structure realised as a UML class diagram. Originally published in [16].

Both the harvesters and the grain wagons are modelled by their physical parameters such as their working/non-working speed, storage capacity and material offload rate. These parameters are specified in the initialisation of the model. The storage point is the out-of-field storage where all material from the field must be transported to in order for the harvest operation to be completed. This too is modelled by its capacity.

The strategy classes define how certain aspects of the harvest operation are executed. In Fig. 1 these strategies are represented by the Route Strategy, Deconflict Strategy and Load Strategy classes.

Route Strategy. A route strategy is responsible for constructing the routes for harvesters. The routes direct the harvester from its location to a point where it will next require a service. A similar approach to the planning of routes for harvesters was also utilised in [4]. In this way the routes for multiple harvesters can be constructed in a consecutive manner.

As already stated, the construction of routes for the harvester and grain wagon are dependent on one another, therefore the route strategy must call functions from the loading strategy to ensure that the harvester is able to be serviced at the end of the route. The route strategies are allowed to produce more than one possible route for the harvester, these are later distinguished by the load strategy as appropriate.

Two route strategies have been implemented within the model: Predefined Route strategy and Greedy Route strategy.

The Predefined Route strategy enables the model to execute coverage plans that have been developed externally, provided they are represented as a sequence of rows to harvest. This strategy receives the assignment of a sequence of rows to a harvester as an input. A route is constructed which navigates the harvester along the sequence of rows, inserting service points where they are needed.

The Greedy Route strategy employs a search algorithm on the field to create a route for the harvester which will end with the harvester being as full as possible and in a position where it can be serviced. An extra constraint is also implemented within the strategy that every row must be harvested in its entirety and that all headland rows must be harvested before work rows.

Deconflict Strategy. A deconflict strategy is responsible for determining if a vehicle can move along its route, or calculating new routes if this is not possible. In the Simple Deconflict strategy a vehicle to reroute is chosen non-deterministically.

A deconflict strategy is responsible for the infield coordination of the vehicles. It is possible that conflicts can arise when a vehicle may block the path of another vehicle. In this case the deconflict strategy is employed to determine what course of action (such as planning a new route, or waiting for the obstruction to pass) is to be taken.

The Simple Deconflict strategy ensures that two vehicles cannot travel towards each other either along the same row or along two adjacent rows.

Load Strategy. A load strategy is responsible for assisting the route strategy to find a location where the harvester can be serviced and for constructing a route for the grain wagon from its current position to the service point and then to the out of field storage.

This is done through three functions of the load strategy that are called by the route strategy: `isDoneExtendingRoute()`, `isRouteServiceable()`, and `finaliseRoute()`.

`isDoneExtendingRoute()` checks if it is possible to extend a harvester's route. A common reason why it would not be possible to extend a harvester's route is if there are no more remaining rows in the field to be harvested, or if the harvester is full.

`finaliseRoute()` modifies a harvester's route to ensure the final position of the harvester is valid. For example if harvesting the full length of the final row of a harvester's route will cause the harvester to exceed its capacity, the route is modified so that a service point is required at some point along the length of the final row.

`isRouteServiceable()` checks that a grain wagon is able to converge on the service point that is required by the harvester's route, for example that there is a previously harvested row adjacent to the service point in which the grain wagon can move.

Four different versions of the load strategy have been developed in the model. These cover the four basic ways in which harvesters are unloaded during grain harvests.

The Single Point Unload version requires the harvester to transport material directly to the out of field storage point without using a grain wagon. It is important that the harvester must avoid the event of becoming full without a navigable path to the out of field storage. This strategy limits the amount of traffic in the field, which could offer benefits when reducing soil compaction.

The Headland Unload version limits the grain wagon to only travelling in the headland areas of the field. The harvester must avoid becoming full in the middle of the field as a grain wagon would not be able to meet it, therefore service points must be coordinated before the harvester becomes full while it is turning in the headland area.

The Infield Static Unload version allows the grain wagons to drive in the working areas of the field in order to meet the harvester. Service points are planned for the last possible moment to ensure that the harvester is full when it unloads.

The Infield Moving Unload version is similar to the Infield Static Unload strategy, however the harvester and the grain wagon are both moving when the load is being passed. As the machines remain in motion it is imperative that the grain wagon is travelling in the same direction as the harvester when they meet at the service point.

The Route, Load and Deconflict strategies are represented in Fig. 1 by their contracts. The various concrete versions of each strategy must conform to these contracts. Figure 2 shows how the various load strategies are realised based on the `ILoadStrategy` class that defines the contract. Whenever the model is executed, a concrete strategy of each kind must be provided to the Execution Engine.

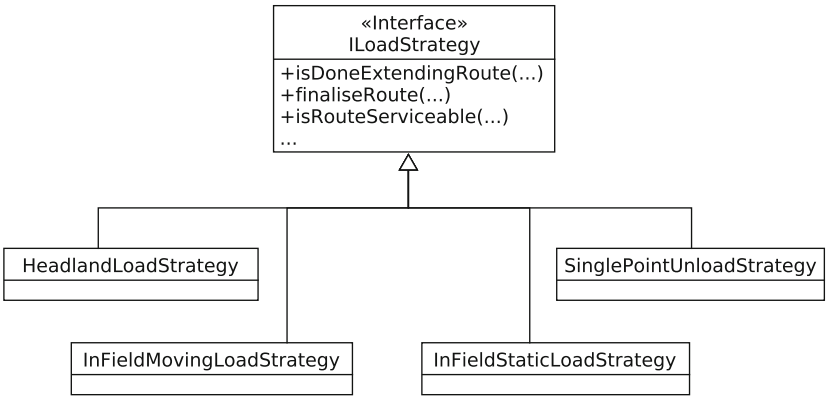


Fig. 2. Load strategy hierarchy realised as a UML class diagram.

Not all versions of a strategy can be used in all situations. In order to cope with this, a notion of *strategy feasibility* has been introduced. The strategy feasibility check is implemented as a function in each of the strategies and invoked at the beginning of model execution in order to check if the field meets the requirements of the strategy configuration. The advantage of this approach is that the feasibility of each version of a strategy is encapsulated in that version itself, so the remaining parts of the model need not be aware of its specific details.

The concrete versions of strategies can be used to model different optimisation algorithms and therefore vary in implementation detail as well as the restrictions they impose on the harvest operation.

3 Model Implementation

The model drives the development of a harvest planning system, which is developed using the Vienna Development Method (VDM) and implemented using code generation. VDM is one of the longest-established formal methods for the development of computer-based systems. This method focuses on the development and analysis of a system model expressed in a formal language.

The strategy pattern is based on object-oriented (OO) features [17], as enabled by the VDM++ formal modelling language [18]. VDM++ is the OO dialect of VDM. Broadly speaking, a VDM++ model consists of a series of definitions for types, functions, operations, etc. The OO features of VDM++ allow for structuring the model into classes and provide standard OO mechanisms such as inheritance.

In addition to allowing for an effective implementation of the strategy pattern, the OO features of VDM++ have other useful benefits, including the ability to add new versions of a strategy that reuse parts of an existing strategy and change only those parts that must be different. Additionally, object-orientation

facilitates modularity and encapsulation which, while not essential to develop the model, make it easier to do so.

There are several reasons for choosing a formal language such as VDM++ over an OO implementation language such as Java or C++. The use of VDM++ promotes a high-level approach that abstracts away details that are of little importance to harvesting operations. The formal semantics underpinning the VDM language allow us to have confidence in the results and that there are no errors in the language and tool that can "contaminate" the result. Additionally, VDM has features that enable us to describe the properties of the model and its functions, and these properties are constantly checked during model execution. For example, in the model the capacity is expressed as a floating point number, which must always be positive and smaller than 1. VDM invariants allow us to attach such a property to the capacity variable in order to ensure that the model never violates this. While that is a simple example, VDM allows us to express any arbitrary property that can be described in terms of first-order logic. Many of the benefits of using VDM cannot be achieved using implementation languages, which operate at a lower level of abstraction. In particular implementation languages must take things such as the underlying hardware platform into account. Use of VDM allows us to focus solely on the development of the strategies, which is our primary concern.

4 Model Execution

In order to execute the model, it is first necessary to configure the harvest operation by loading both the field and the resources, i.e. the State, and also one of each class of strategy to guide the Execution Engine during the simulation. Once this is done, the model is executed and whenever the Execution Engine reaches a point where it needs to make a decision that depends on a strategy, it will consult whatever strategy it has loaded and the output of the strategy will be used to further progress execution of the model. As an example, in Fig. 3, the Execution Engine needs to know which vehicles are movable at a given point in time. One particular version of the strategy may allow the harvesters to move because they can offload in the work rows. Another version may not allow the harvesters to move because they can only offload in the headlands and they cannot fully harvest the next work row.[2] In this way, different versions of a strategy lead to different outcomes in the model.

One of the key features of the model is the ability to explore strategy combinations and how their interactions affect the performance of the harvest operation. One way to do this is by fixing two kinds of strategies and varying the remainder (for example, load strategies) thus investigating how a particular aspect of optimisation affects the overall harvest operation. Conversely, if external restrictions dictate the use of a particular strategy, then the other strategies may be manipulated to find the best solution within the restrictions. For a small number of

[2] In both of these examples, the route strategy consults the load strategy as part of its calculation of movable vehicles.

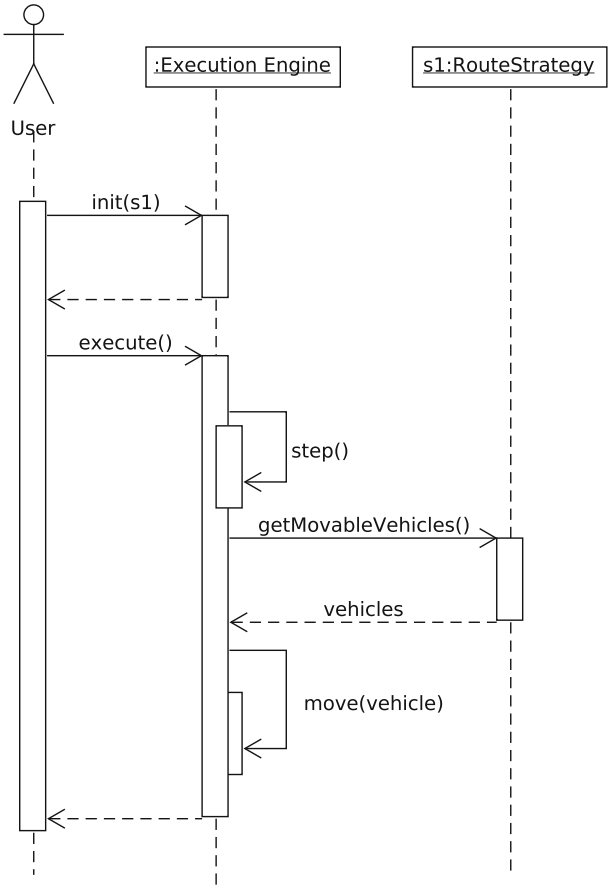


Fig. 3. Strategy dispatching realised as a UML sequence diagram. Originally published in [16].

strategies, testing the different scenarios of interest can be done with manually written tests. However, when the number of scenarios to be tested is large then an automated combinatorial testing feature for VDM can be used to concisely specify the various combinations and automatically generate and execute the corresponding tests [19].

4.1 Simulation Visualisation

As part of model execution, a log of all the important events in the harvest operation is produced. Logged events include vehicle movement, harvesting of a row, passing load between harvesters and grain wagons, etc. Once execution is completed, this log can be inspected in order to get a full understanding of the

harvest operation outcome. This log can also be seen as a harvest plan since it contains detailed instructions of when and where the different vehicles must go.

In order to better understand what occurred during the simulation, the log can also be analysed. However, as manual inspection of the log is difficult, a proof-of-concept visualization tool was developed to analyse the log and replay the simulation as shown in Fig. 4. The figure shows a representation of the field partitioned into work rows and headlands. The black square represents the harvester, the circle represents the grain wagon and the square at the bottom represents the storage point. As the log is processed, the visualiser displays an animation of the vehicles moving along the field.

Fig. 4. Simulation visualisation. Originally published in [16].

5 Results

This section demonstrates the approach by reporting results of executing various simulations with the model in order to explore the interactions between all possible combinations of the strategies described in Sect. 2.1. Every execution was performed with the same resources and on the same field. The focus is not

on changing the parameters of the simulation such as number of harvesters or harvester capacity but in changing the strategy versions used in each simulation.

The simulations were carried out on a representation of a real field located in the vicinity of the Research Center at Foulum, Denmark (56°29′N, 9°35′E). The yield of the field is simulated and is lower for headland rows than for working rows, as is typical in real fields (due to excess soil damage, lower nutrients, etc.). The yield is further constrained such that a complete lap of the field can be made without exceeding the harvester capacity, and no single working row can exceed the capacity of the harvester. The field, partitioned into rows, is shown in Fig. 5.

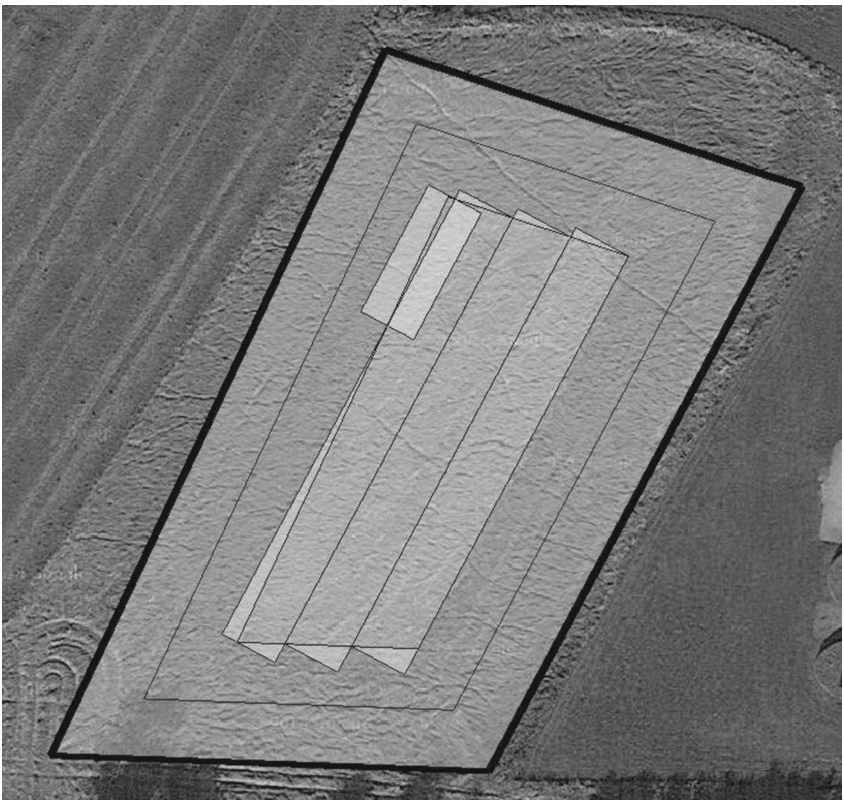


Fig. 5. Agro Park field. Originally published in [16].

The results of the simulations are summarised in Table 1. Each row in the table represents a particular simulation, indexed by the *Sim.* (Simulation) column. The *Route* and *Load* columns identify the combination of strategies used in each particular simulation (the same deconflict strategy – Simple Deconflict – is used for all simulations). The *Op. Time* (Operational Time) column reports the

duration of the harvest operation in seconds and serves as an indication of how well a combination of strategies performs. Finally the *Exec. Time* (Execution Time) column reports the actual, physical time in seconds it takes to execute the simulation.

The simulation was executed using a Java 7 code generated version of the model on a Fujitsu LIFEBOOK U772 laptop with a 1.7 GHz Intel Core i5 processor and 8 GB of memory running a Windows 7 Professional Edition operating system.

Table 1. Results summary. Originally published in [16].

Sim	Route	Load	Op. time [s]	Exec. time [s]
1	Greedy	Headlands	425.558	12.619
2	Predefined	Headlands	497.38	13.417
3	Greedy	In field static	420.694	12.319
4	Predefined	In field static	463.484	13.912
5	Greedy	In field moving	410.298	7.056
6	Predefined	In field moving	446.854	7.25
7	Greedy	Single point	679.498	26.977
8	Predefined	Single point	623.347	4.421

6 Discussion

Table 1 shows that for the field subject to analysis, for most of the unloading strategies, the *Greedy Route* strategy produces a better solution, than the *Predefined Route* strategy as indicated by the operational time. This is due to the harvester's route used as an input for the *Predefined Route* strategy being developed as a coverage plan that ignores the coordination of the grain wagons. As the *Greedy Route* strategy was able to enquire the constraints of the unloading strategy while developing the harvester's route, the final solution is more integrated and allows for more efficient operations. This indicates that it may be advantageous to use optimisation approaches that consider both harvesters and grain wagons when developing routes.

The *Infield Moving Unloading* strategy offers the best operational times for both of the routing strategies. This unloading strategy is likely to offer the best solution as it allows the harvester to be completely full when it offloads and does not require the harvester to stop. It is also worth noting that the model allows this hypothesis to be further confirmed by adding additional route strategies and checking the resulting operational times.

In terms of actual execution times, most combinations yield similar results for *Greedy* and *Predefined* strategies. The exception is for the *Single Point Unload* strategy, where the *Greedy* version has a significantly higher execution time. This is mostly due to the fact that many more routes have to be computed for this

particular combination, which makes it significantly slower than its *Predefined Route* counterpart.

The field used for these simulations is small, especially in terms of the numbers of rows and headland laps. Indeed, when the model was under initial development, small fields were preferred as they allowed for quick execution of model simulation, which enabled fast iterations of model development. However, as the model stabilised and we began to apply the system for the harvest planning of larger fields, we experienced significant performance issues.

For larger fields (ten rows or more), the performance of the system was unacceptably slow. The primary culprit for the poor performance was the data representation of the field and the algorithms used to implement common operations on the field (such as shortest path calculation). One of the reasons for these inefficient implementations was the use of VDM itself. As a formal modelling language, VDM and its associated tools are more concerned with semantic fidelity and validity of analyses than with simulation performance.

To address this issue, the field representation in the model was replaced with a handwritten Java implementation. The Overture VDM-Java bridge [20] was used to connect this implementation to the model, and a new mechanism called the *delegate* was introduced to ensure seamless realisation and integration of the final system. The introduction of the Java component led to performance gains of 3000% [21] that effectively addressed the performance issues of the previous version of the model and helped us achieve acceptable system performance for much larger fields.

7 Live Planning

The harvest planning system as described so far uses *offline planning* techniques to analyse harvest operations. Essentially, this means that planning is only done once, and that the system takes no measures against unforeseen scenarios that may necessitate re-planning over the course of the harvest operation. *Live planning* techniques, on the other hand, continuously analyse data reported by the vehicles (positions, bin levels etc.) and try to optimise the harvest plan by assigning more efficient routes to the vehicles. A live version of the system can therefore be seen as a tool that serves to guide the operators of the vehicles throughout the harvest operation.

A live system is currently under development, and we expect to use it in a realistic harvest setting in the near future. The live system will run on a server that receives live data from the vehicles and uses this data to further improve the harvest plan. Once new routes have been calculated, these will be assigned to the vehicles, which connect to the server via light-weight clients. A light-weight client is responsible for communicating live data to the server and provide the operator of the vehicle with information about the route that the vehicle is currently assigned. The structure of the live system is visualised in Fig. 6.

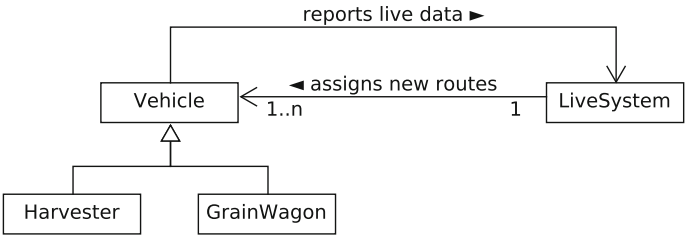


Fig. 6. The structure of the live system realised as a UML class diagram.

As a first step, the live planning system computes initial routes for each vehicle, which is similar to what the offline planning system already does. Based on the initial routes, the vehicles start to harvest the field, and as the harvest progresses, the vehicles report live data to the server. If the data reported by the vehicles indicates a need for re-planning, then the live system responds by calculating and assigning new routes to the vehicles. This step is repeated until the harvest operation has completed.

Several situations may occur which necessitate re-planning. For example, a harvester may report a bin level that is smaller than what is expected by the system. In that case the system may decide to extend the harvester's route to compensate for the smaller bin level. However, changing the harvester's route may necessitate re-calculations of unload points, hence also affecting the routes of other vehicles. As another example, a vehicle may encounter an unexpected obstacle in the field that prevents it from following the route it has been assigned. Similar to the previous example, such situations necessitate re-planning which may affect the other vehicles that participate in the harvest operation.

The harvest planning system is only a viable solution if using the system leads to better harvest results. To show this, the results obtained using the live system will be compared to those obtained using traditional harvest approaches, i.e. before the harvest planning system was used. When the live system has been used in a realistic harvest setting it will be possible to obtain quantitative evidence that shows if the system achieves better results. Demonstration of this is crucial in order to convince farmers to start using the system.

8 Conclusions and Future Work

In this paper, we have presented the model-based development of a harvest planning system. We have shown how the strategy pattern enables the application of different optimisation algorithms to different phases of the harvest planning system. Further, we have shown how such algorithms can be easily varied across multiple simulations, enabling a swift and comprehensive exploration of the design space.

We have shown an example application of our system to develop plans for a small real field in Denmark. As the size of the fields increases, the performance of the model was greatly degraded. This was dealt with by replacing a

portion of the model with a handwritten Java component that provides efficient implementations of the more computationally-intensive operations in the model.

The next step in our work is to validate the plans produced by the system by applying them to a real harvest and verifying if the results are better than those obtained with traditional harvest planning approaches. Towards that end, work has begun on a live planning system that will assist vehicle operators in following a harvest plan. This system will also take advantage of live data to adapt and improve the plan on-the-fly.

Acknowledgements. A previous version of this paper was presented at the SIMULTECH 2016 conference [16]. The work described in this paper was partially carried out in the context of the Danish High Technology Foundation research project Off-line and on-line logistics planning of harvesting processes. We would like to thank all our colleagues on the project for their valuable contributions and feedback, particularly Peter Gorm Larsen, Claus Grøn Sørensen, Dionysis Bochtis and Morten Bilde. We also thank Kun Zhou for his assistance with the harvest visualisation.

References

1. Spekken, M., de Bruin, S.: Optimized routing on agricultural fields by minimizing maneuvering and servicing time. Precis. Agric. **14**, 224–244 (2013)
2. Edwards, G., Christiansen, M.P., Bochtis, D.D., Sørensen, C.G.: A test platform for planned field operations using LEGO mindstorms NXT. Robotics **2**, 203–216 (2013)
3. Jensen, M.A.F., Bochtis, D.D., Sørensen, C.G., Blas, M.R., Lykkegaard, K.L.: In-field and inter-field path planning for agricultural transport units. Comput. Indus. Eng. **63**, 1054–1061 (2012)
4. Oksanen, T., Visala, A.: Coverage path planning algorithms for agricultural field machines. J. Field Robot. **26**, 651–668 (2009)
5. Edwards, G., Jensen, M.A.F., Bochtis, D.D.: Coverage planning for capacitated field operations under spatial variability. Int. J. Sustain. Agric. Manag. Inf. **1**, 120–129 (2015)
6. Bochtis, D.D., Sørensen, C.G.: The vehicle routing problem in field logistics part I. Biosyst. Eng. **104**, 447–457 (2009)
7. Jensen, M.A.F.: Operations planning for agricultural machinery under capacity constraints. Ph.D. thesis, Aarhus University (2014)
8. Scheuren, S., Stiene, S., Hartanto, R., Hertzberg, J., Reinecke, M.: Spatio-temporally constrained planning for cooperative vehicles in a harvesting scenario. KI-Künstliche Intelligenz **27**, 341–346 (2013)
9. Jin, J., Tang, L.: Optimal coverage path planning for arable farming on 2d surfaces. Trans. ASABE **53**, 283 (2010)
10. Zandonadi, R.S.: Computational tools for improving route planning in agricultural field operations. Ph.D. thesis, University of Kentucky (2012)
11. Hameed, I., Bochtis, D.D., Sørensen, C.G., Jensen, A.L., Larsen, R.: Optimized driving direction based on a three-dimensional field representation. Comput. Electron. Agric. **91**, 145–153 (2013)
12. Tullberg, J.: Tillage, traffic and sustainability – a challenge for ISTRO. Soil Tillage Res. **111**, 26–32 (2010)

13. Fitzgerald, J., Larsen, P.G., Verhoef, M. (eds.): Collaborative Design for Embedded Systems – Co-modelling and Co-simulation. Springer, Heidelberg (2014)
14. Broenink, J.F., Fitzgerald, J., Gamble, C., Ingram, C., Mader, A., Marincic, J., Ni, Y., Pierce, K., Zhang, X.: Methodological guidelines 3. Technical report, The DESTECS Project (INFSO-ICT-248134) (2012)
15. Gamma, E., Helm, R., Johnson, R., Vlissides, R.: Design Patterns. Elements of Reusable Object-Oriented Software. Addison-Wesley Publishing Company, Reading (1995)
16. Couto, L.D., Tran-Jørgensen, P.W.V., Edwards, G.T.C.: Combining harvesting operation optimisations using strategy-based simulation. In: Proceedings of the 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications - Volume 1: SIMULTECH, pp. 25–32 (2016)
17. Meyer, B.: Object-Oriented Software Construction. Prentice-Hall International, Upper Saddle River (1988)
18. Fitzgerald, J., Larsen, P.G., Mukherjee, P., Plat, N., Verhoef, M.: Validated Designs for Object-Oriented Systems. Springer, New York (2005)
19. Larsen, P.G., Lausdahl, K., Battle, N.: Combinatorial testing for VDM. In: Proceedings of the 2010 8th IEEE International Conference on Software Engineering and Formal Methods, SEFM 2010, Washington, DC, USA, pp. 278–285. IEEE Computer Society (2010). ISBN 978-0-7695-4153-2
20. Nielsen, C.B., Lausdahl, K., Larsen, P.G.: Combining VDM with executable code. In: Derrick, J., Fitzgerald, J., Gnesi, S., Khurshid, S., Leuschel, M., Reeves, S., Riccobene, E. (eds.) Abstract State Machines, Alloy, B, VDM, and Z. Lecture Notes in Computer Science, vol. 7316, pp. 266–279. Springer, Heidelberg (2012)
21. Couto, L.D., Tran-Jørgensen, P.W.V.: Integrating real system components in model-based development. Submitted to the 33rd ACM/SIGAPP Symposium on Applied Computing (SAC 2018)

Cluster Performance Simulation for Spark Deployment Planning, Evaluation and Optimization

Qian Chen[1(✉)], Kebing Wang[1], Zhaojuan Bian[1], Illia Cremer[2], Gen Xu[1], and Yejun Guo[1]

[1] Software and Service Group, Intel Corporation, Shang Hai, China
{charles.chen, kebing.wang, bianny.bian, gen.xu, yejun.a.guo}@intel.com
[2] Software and Service Group, Intel Corporation, Nantes, France
illia.cremer@intel.com

Abstract. As the most active project in the Hadoop ecosystem these days [1], Spark is a fast and general purpose engine for large-scale data processing. Spark runs programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk [2]. However, Spark performance is impacted by many factors especially memory and JVM related, which makes capacity planning and tuning for Spark clusters extremely difficult. Current estimation based solution are highly dependent on experience which are trial-and-error and far from efficient and accurate. Here, we propose a novel Spark simulator based on CSMethod [3], extension with a fine-grained multi-layered memory subsystem, well suitable for this scenario. The whole Spark application execution life cycle is simulated, hardware activities derived from software operations are dynamically mapped onto architecture models for processors, storage, and network devices. Experimental results with several popular micro benchmarks and a real case IoT workloads demonstrate that our Spark Simulator achieves high accuracy with an average error rate below 7%, with light weight computing resource. Case studies are also demonstrated to show the simulator's capability.

Keywords: Spark simulation · Cluster simulation · Performance modelling · Memory modelling · In-Memory computing · Big data · Capacity planning · IoT

1 Introduction

Spark is an open-source data analytics cluster computing framework, which promises performance up to 100 times faster than Hadoop MapReduce for certain applications [4]. It provides primitives for in-memory cluster computing that allows user programs to load data into a cluster's memory and query it repeatedly [5]. In Spark, data is abstractly represented by RDD. Users can explicitly cache an RDD in memory across machines and reuse it in multiple MapReduce like parallel operations [6]. Spark became an Apache top-level project in February 2014 [7].

In Spark performance tuning, memory related tuning should be a high priority. Spark provides a wide range of hardware and software parameters to control the

M.S. Obaidat et al. (eds.), *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing 676,
https://doi.org/10.1007/978-3-319-69832-8_3

memory behaviour. Since complex interactions exist between these parameters, it is very difficult to find an optimized parameters configuration that would maximize the Spark cluster performance.

Traditional Cluster design and deployment decision are experience or measurement based, which can't meet Spark cluster deployment criterions very well. Due to the very new nature of Spark, very few users can take sound and accurate decisions based on experience. On the other hand, upon cluster availability, measurement based optimization is extremely time consuming and can be easily interrupted by random environment factors like disk or network interface card (NIC) failures.

Simulation based cluster analysis in general is a much more reliable approach to obtain systematic optimization solutions. Among the various simulation methods proposed [8–11], CSMethod [3] is a fast and accurate cluster simulation method which employs a layered and configurable architecture to simulate Big Data clusters on standard client computers (desktop or laptop).

The Spark workflow, especially the DAG abstraction, is very different from the Hadoop MapReduce workflow. In addition, current CSMethod based MapReduce model's memory subsystem is too coarse to meet accuracy requirements for Spark simulation. To fill these gaps, this paper proposes a new simulation framework which is based on and extending CSMethod. All performance intensive Spark parameters and workflow are modeled for fast and accurate performance prediction with a fine-grained multi-layer memory subsystem.

The whole Spark cluster software stack is abstracted and simulated at functional level, including computing, communications and dataset access. Software functions are dynamically mapped onto hardware components. The timing of hardware components (storage, network, memory and CPU) is modeled according to payload and activities as perceived by software. A low overhead discrete-event simulation engine enables fast simulation speed and good scalability. The Spark simulator accepts Spark applications with input dataset information and cluster configurations then simulates the performance behaviour of the Spark application. The cluster configuration includes the software stack configuration and the hardware components configuration.

The following key contributions are presented in this paper:

- We propose a new framework to simulate the whole performance intensive Spark workflow, including: DAG generation; RDD input fetch, transfer, shuffle and block management; Spill and HDFS access.
- We describe a fine-grained multi-layer memory performance model which simulates the memory behaviour of Spark, JVM, OS and H/W layers with high accuracy.
- We demonstrate a simulation based Spark cluster deployment planning evaluation and optimization approach with high accuracy (>93%) and low computing cost.

The rest of this paper is organized as follows. Section 2 presents the proposed Spark simulator in details. The experimental environment set up and the workload are then introduced in Sect. 3. Section 4 illustrates the evaluation results and its analyses. Real case IoT Spark cluster deployment planning and Memory related Spark performance tuning case studies are then presented in details in Sects. 5 and 6. Section 7 overviews related work. A summary and future work thoughts are described in the final section.

2 Spark Simulation Framework Architecture

In this section, we introduce the proposed Spark simulation framework in details.

2.1 Spark Behaviour Model

The proposed Spark model was developed using Intel®CoFluent™ Studio [12] which provides an easy to use graphical modeling tool in a System-C simulation environment.

Simulation speed of our performance simulator is faster than general simulators because we abstract actual computation down to time estimation.

1. The software behaviours (data flow) are divided into several basic operations such as compression, serialization, sorting, partition, match, mathematical computation, hash, shuffle, file system/memory access, etc. These basic operations are then dynamically mapped to hardware timing models which would return the timing of these operations.
2. The hardware models are implemented as a global performance library. The timing and utilization of hardware resources like CPU, memory, disk, network, and cluster topology are modeled. The modeling principle is CSMethod which is described in another paper [3]. To provide a fast understanding of CSMethod, here we give a short example of CPU computing time estimating modelling.

$$\tau = \alpha \times \beta \times \gamma \times \delta \div \varepsilon \tag{1}$$

Where τ is computing time of a software operation like java serialization; α is CPU Cost which is a function of CPI (Clocks Per Instruction) i.e. $\alpha = f(CPI)$; β is data set size; γ is performance indicator, for example, if a processor is running at 1.6 GHZ out of maximum frequency 2.7 GHZ then $\gamma = 2.7 \div 1.6$, δ is current running thread count which is dynamically modeled and tracked by simulator and, ε is CPU core count.

The top level of our Spark behaviour model is shown in Fig. 1, which includes: a Master, a Network, Slaves, Clients and AppMasters. Clients submit jobs to the cluster and act as workload generators. The Master takes the resource management role. The AppMaster analyses the job configuration and generates a DAG of tasks to be executed by slave nodes in the cluster. The Network connects all the components logically and simulates the Cluster network topology, bandwidth and latency. The Clock model synchronizes the timing between all the logic blocks. The Slaves receive tasks generated by the AppMaster and launch the TaskRunner to execute the tasks with resources provided by the NodeResourceManager in themselves.

As shown in Fig. 2 the TaskRunner simulates the Spark task workflow behaviour, including: fetchInput, RDD transfer, Shuffle and result computing.

Different types of task inputs can be fetched by the FetchInput module: HadoopRDD, cached RDD, or shuffle RDD tasks. The remote shuffle data are copied by the Copier and the Router which are connected to the network module to simulate shuffle behaviour. Fetched RDD blocks are transformed by specific RDD operations in the RDDTransform module. Depending on the specified Spark task type, the transformed RDD block can be used to form the result output or the shuffle output.

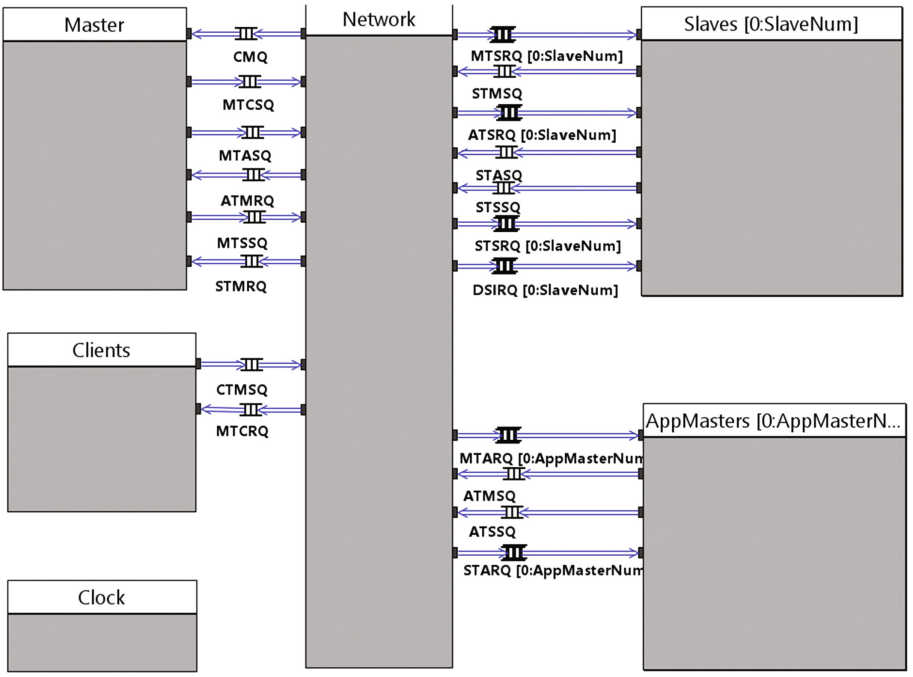


Fig. 1. Top abstraction level of the Spark model.

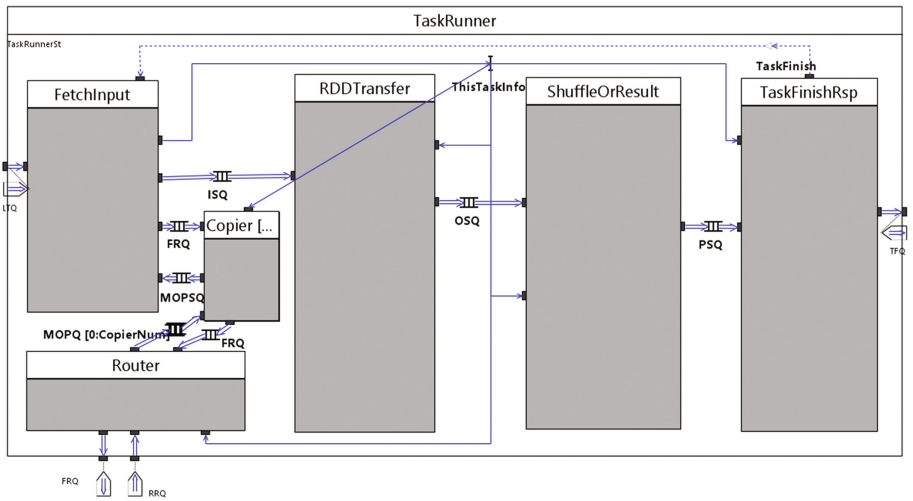


Fig. 2. Middle abstraction level model of the Spark task executor.

The result output is generated by the ResultTask module which computes the final result and writes it to HDFS. The shuffle output is generated by the ShuffleMapTask module that partitions the output in hash keys and then dispatches the output to specific shuffle output files.

Finally the 'task finish' signal and the task performance metrics are committed to the scheduler, then the TaskRunner module waits for the next task to be dispatched to it. Performance intensive software functions like compress, decompress, serialize, de-serialize, sort, hash operations are modeled within the TaskRunner module.

2.2 Memory Model

The RDD Block manager and the performance library are used by the TaskRunner module to simulate dataflow events (RDD block read/write, JVM/OS memory apply/free, disk read/write, CPU apply/free, network transfer, HDFS read/write) and to generate the timing information. During the whole simulation cycle the cluster hardware resource usage is tracked and updated dynamically by the performance library.

In order to obtain high accuracy, the Spark simulation memory model has been implemented in 4 different layers: Spark, JVM, OS and physical memory. The last three layers are modeled as a global memory performance library as shown in Fig. 3, and was called by the Spark layer to simulate RDD block management behaviour.

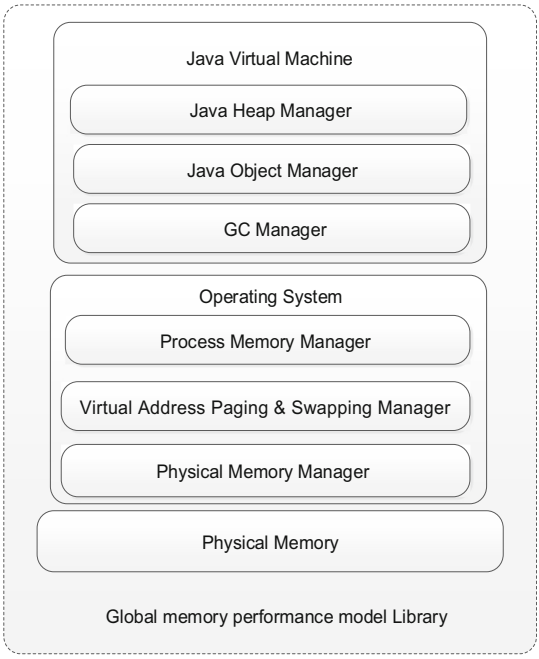


Fig. 3. Structure of multiple layered memory model.

- Spark software stack layer: Spark RDD block management behaviour are modeled by simulating RDD block put, get and cache operations.
- JVM layer: the JVM heap space capacity limits, GC triggering mechanism and object management behaviour are modeled, so that when JVM heap doesn't have enough free space to hold new object then GC happens.
- OS layer: the virtual space, paging, swapping slab and disk file cache/buffer behaviour are modeled. The file (usually on disk) access requests are cached or buffered by OS managed free system memory.
- Physical memory layer: the physical memory bandwidth latency and capacity limits are modeled by keeping track of concurrent memory accesses.

The hierarchy of this memory model is similar to real systems, each level is itself a class and has respective behaviours and can be inherited. The simulation granularity is configurable to achieve the simulation trade-off between accuracy and speed, for example swapping operation could be done per page or per block.

This memory model simulates the full system memory behaviour within a single process in a standard personal computer with timely response.

2.3 Simulator User Input

The input of the simulator is composed of the Spark S/W stack configuration, the H/W components configuration and the Spark application/job abstraction.

Table 1. Cluster hardware and software settings.

Cluster	Node number and network topology
Processor	Processor type, core count, thread count and frequency
Storage	Storage count and type: SSD or HDD
Memory	Memory type and capacity
Network	NIC count and bandwidth 10 or 1 GBit/s

The cluster hardware components configuration is listed in Table 1. While the Spark software stack configuration is listed in Table 2.

Table 2. Spark JVM OS parameters.

Level	Modeled Software Parameters
Spark	Spark.executor.memory
	Spark.default.parallelism
	Spark.storage.memoryFraction
	Spark.shuffle.compress
	Spark.shuffle.spill.compress
	Spark.rdd.compress
	Spark.io.compression.codec

(*continued*)

Table 2. (*continued*)

Level	Modeled Software Parameters
	Spark.io.compression.snappy.block.size
	Spark.reducer.maxMbInFlight
	Spark.shuffle.consolidateFiles
	Spark.shuffle.file.buffer.kb
	Spark.shuffle.spill
	Spark.closure.serializer
	Spark.kryo.referenceTracking
	Spark.kryoserializer.buffer.mb
	Spark.shuffle.memoryFraction
	SchedulerReviveInterval
	Akka threads number
YARN	Yarn.scheduler.minimum-allocation-mb
	Yarn.scheduler.increment-allocation-mb
	Yarn.scheduler.maximum-allocation-mb
	Yarn.scheduler.minimum-allocation-vcores
	Yarn.scheduler.increment-allocation-vcores
	Yarn.scheduler.maximum-allocation-vcores
HDFS	Dfs.block.size
JVM	Heap size
	Young generation ratio
	EdenSurvRatio
	GC drop ratio
OS	Memory flush ratio
	Memory dirty ratio
	Memory flush interval
	Transparent huge page

Model abstraction is defined from the following aspects: RDD information: size, partition number, and storage location (HDFS, shuffle and memory cache). Operation information: operation type (shuffle or map) and operation CPU cost.

2.4 Simulator Output

Timelines, charts and console output windows provided by the Intel CoFluent Studio development toolkit are used to visualize metrics. Many other metrics extracted from output result files are also observed using spreadsheets.

3 Experimental Setup

This section describes the configuration of our experimental setup. It is followed by a presentation of the benchmarks used for the evaluation of the model.

3.1 Experiment Cluster

Table 3 lists the target cluster hardware components and the software stack elements used for our baseline experiments. This setup is representative of mainstream data-center configurations used for Big Data processing.

Table 3. Cluster hardware and software settings.

Cluster	5 Nodes, connected by one rack switch 4 slave worker nodes 1 master node
Processor	Intel® Xeon® E5-2697 v2, 24 cores per node with HT disabled
Disk	Direct Attached Storage, 5 × 600 GB SSD per node, 1 drive for OS, 4 drive for Spark S/W stack
Memory	128 GB, 2 channel DDR3-1333 per node
Network	10 Gbit/s Ethernet
OS	RedHat6.4
Java	1.7.0_67
Spark	Spark 1.2
Platform	CDH5.2

3.2 Workload Description

Three workloads are used to conduct the experiments. Widely used in machine learning, K-Means clustering is a method of vector quantization, popular for cluster analysis in data mining. As an iterative application Spark K-Means is often used as a typical application to show Spark advantage. PageRank is another good example of a more complex algorithm with multiple stages of map and reduce iterations. It benefits from Spark's in-memory caching mechanism with multiple iterations over the same data. SparkTC is an implementation of transitive closure. It can be thought of as establishing a data structure that makes it possible to solve reachability questions [13]. The configuration of these three workloads are shown in Table 4.

Table 4. Experimental workload baseline configurations.

Parameters	Value
K-Means input data set size in GB	40/80/160
K-Means dimensions	30
K-Means iteration number	5
K-Means cluster number	1024
PageRank input data set size in GB	11/22/40
PageRank iteration number	5
SparkTC edges	200
SparkTC vertices	100/200/400

4 Evaluation and Analysis

In addition to above micro-benchmark, we have also validated our simulator with 2 machine learning algorithms: SVM, ALS and an IoT real case usage scenario, all with error rate less than 7%. As the limitation of this paper, this section only describes the micro-benchmark validation in detail.

The Spark simulator accepts 33 parameters for each workload simulation, but we only choose several parameters to do performance trend study, which are related to the system performance bottleneck. Only the most sensitive parameters are scaled while the other parameters are set as default.

4.1 Baseline Validation

Three different workload input data sizes were used to illustrate the accuracy of our simulator. The detailed workload input parameters are shown in Table 4.

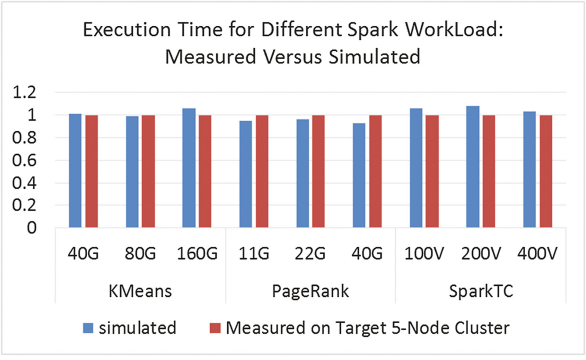


Fig. 4. Measurement Vs. simulation of Spark performance.

Figure 4 shows normalized Spark execution times as measured on the experimental cluster and as predicted by the simulator. As we can see, the simulation results are always very close to the real hardware measurements, the average error rate is 4.5%.

4.2 Memory Model Accuracy Analysis

The simulation accuracy of memory related parameter is evaluated at three different system levels: Spark, JVM and OS. As the model at higher system level are based on the lower ones, the simulation accuracy of higher level are lower than that of the lower one. As shown in Fig. 5, all average error rate are less than 7%.

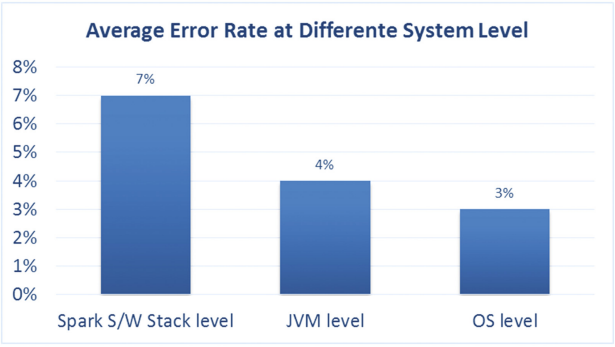


Fig. 5. Simulation accuracy of memory model.

4.3 Software and Hardware Parameters Scalability Analysis

The scalability analysis has been extended to all software and hardware parameters supported by the framework which are list in Tables 1 and 2. It shows that the average error rate between actual performance and simulated performance is within 6% regardless of the type of the software parameter being changed. For hardware parameter scaling, the average error rate is within 5%.

As software parameter scaling examples will be descript in detail in Sect. 5, here we focus on a processor scaling example to show the hardware parameter scaling ability of our Spark model. The computing intensive K-Means workload was selected for this evaluation.

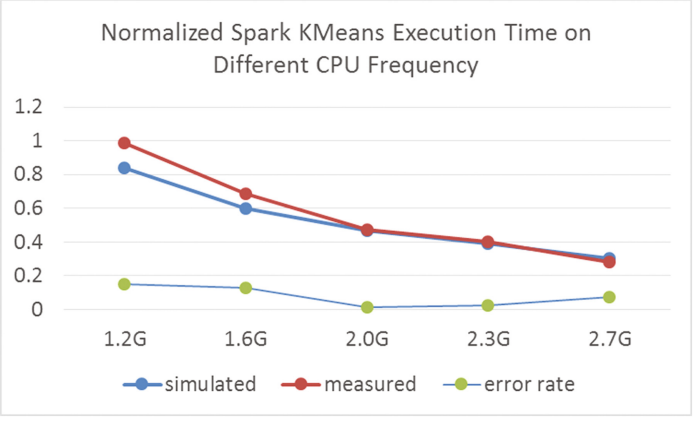


Fig. 6. Normalized execution time of CPU Frequency Scaling.

Figure 6 shows the CPU frequency scaling for the K-Means workload. Higher CPU frequencies improve the processing performance and reduce the workload execution time. The simulated performance has the same trend as the measured performance and the average error rate is 7.7%.

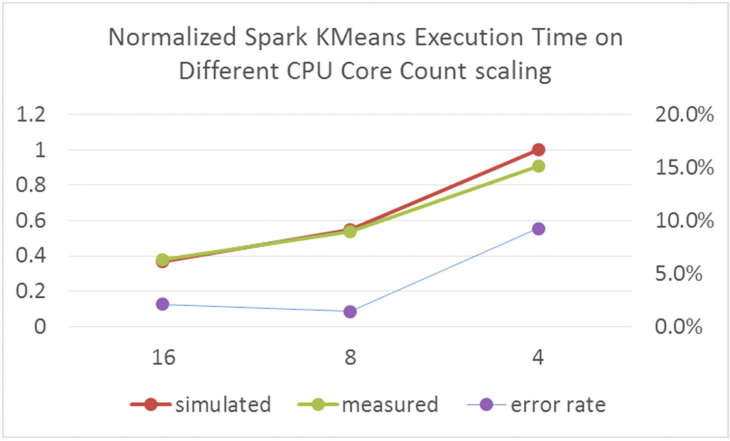


Fig. 7. Normalized CPU core count scaling.

Figure 7 shows the CPU core count scaling for the K-Means workload. More CPU cores reduce the workload execution time. Simulated and measured performance have the same trend with an average error rate of 4.2%.

4.4 Simulation Speed

All simulations are running on a standard desktop equipped Intel(R) Core i7-5960 CPU and 16 GB DDR memory. For different benchmarks and configurations, the native execution time on experiment cluster ranges from 10 min ~ 30 min. To predict the native execution time, the simulator would cost 15 min to 4 h.

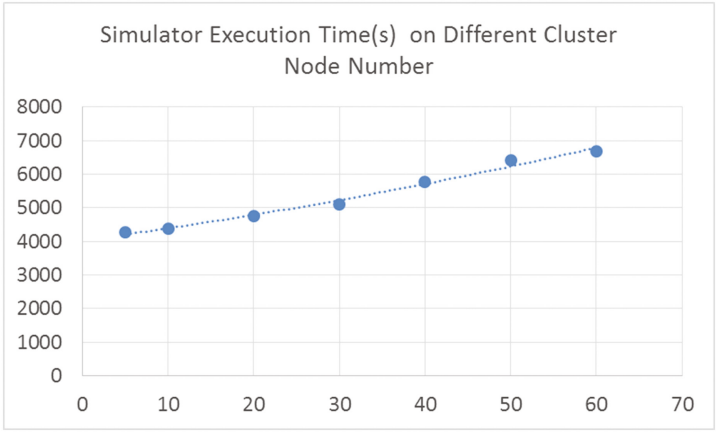


Fig. 8. Simulator execution time of 50 GB dataset for various node counts.

Figure 8 shows the actual simulation processing time for a 50 GB data set processed by the Spark PageRank workload. The cluster size is scaled from 5 to 60 nodes. The simulation processing time ranges from 1 to 2 h. This simulation speed is slower than the lighting fast in memory computing engine: Spark, but still acceptable for cluster deployment planning evaluation and optimization.

5 Case Study 1: Spark Cluster Deployment Planning

Cardiac arrhythmia is broad term describing dozens if not hundreds of conditions which range from benign to life-threatening. Patients with various cardiac arrhythmia conditions often wear Cardiac Event Recorders (CERs). Cardiac Event Recorders are portable devices which record the heart's electrical activity while the patient is going about their normal routine. CERs, and many other medical devices can automatically upload data to data center via the patient's smartphone [14]. The solution system architecture shown in Fig. 9.

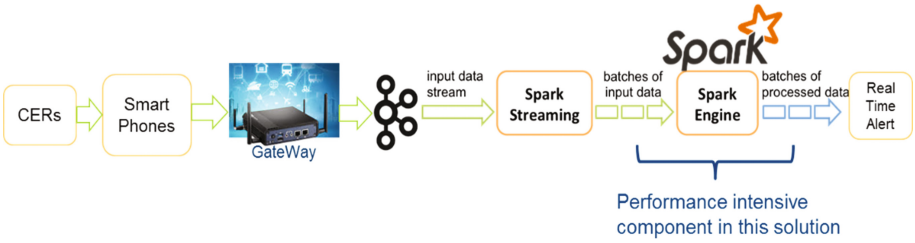


Fig. 9. Cardiac IoT real time monitoring and alerting solution architecture.

This is a high-level description of a BigData analytics system which can utilize machine learning to perform real-time alerting and also near real-time reporting for patients and medical professionals. Kafka is used as a buffer before data is written to HDFS and also ingested by Spark Streaming which is used to perform real-time analytics and alerting contacts when conditions are identified such as ventricular fibrillation. Patients and medical professions are able to analyze and visualize data on a website served by Impala.

In this solution spark engine cluster is the system performance bottleneck, the engine should be powerful enough to process streaming data in real time. Streaming jobs will execute every few seconds and should operate on data generated immediately preceding the window. In short, the maximum time data waits to be processed should be less than or equal to the window.

Assuming 300 fields with naive serialization, each record should be about 300 * 4 bytes which is 1.2 KB. Four bytes is assumed since these outputs are typically either small integer or floating point values. Another data point is that cardiac arrhythmia affects 2% to 3% of the population or about 10 million people in the United States.

One event per minute means we'll be ingesting about 200 MB per second (10 million people * 1.2 KB per event/60 s) or 12 GB per minute.

To figure out the capability of our 4-node baseline cluster in processing this streaming task through performance simulation. As shown in Fig. 10 with the data ingestion rate increase from 0.6 million to 3.2 million kilo record per second, the spark cluster will failed to process the data in real time while ingestion rate is larger than 2.4 million kilo record per second.

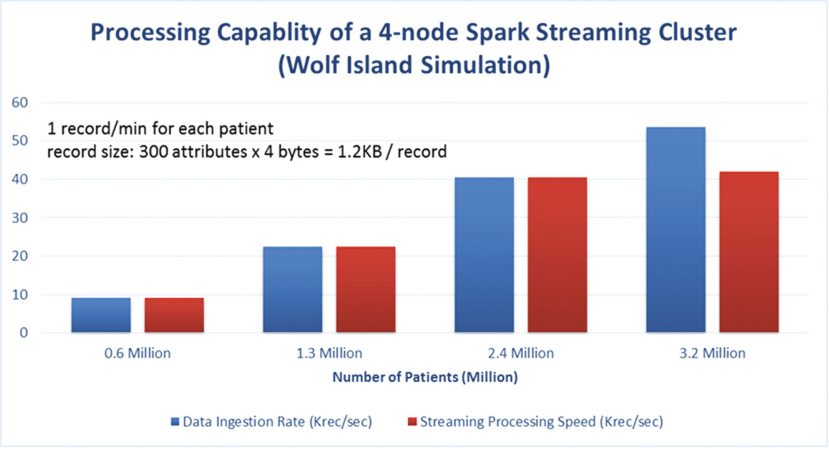


Fig. 10. Baseline cluster processing capability.

To support more CERs devices spark cluster needs to be scaled up and vice versa. Simulation based spark performance scaling for cluster deployment planning will be demonstrated in the following section.

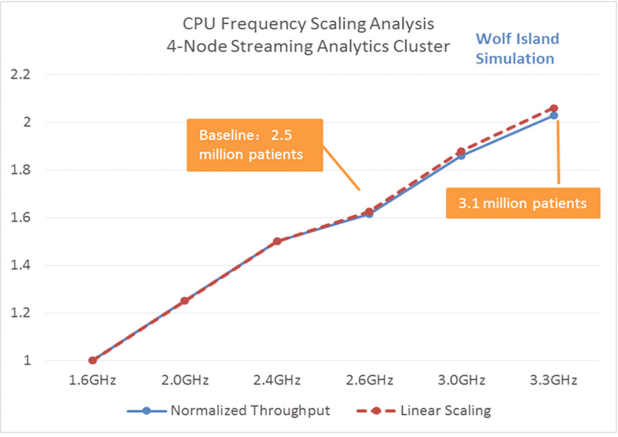


Fig. 11. CPU frequency scaling result.

Figure 11 represents the capability of baseline cluster scaling with CPU operating frequency. As predicted by simulator, the cluster capability will increase to support 3.1 million patients with frequency increased to 3.3 GHZ from baseline value of 2.6 GHZ. As 3.3 GHZ is the maximum CPU frequency, if more than 3.1 million patients need to be support, then we have to scaling other cluster configurations.

Simulated result of spark cluster capability scaling with core count on each server node is shown in Fig. 12, with core count increased to 28 per node the scaled cluster could support about 4.25 million patients. The normalized throughput scaling trends is very close to linear scaling (red line), which means current workload is completely CPU intensive. In this case more CPU core or higher CPU frequency could scale up the overall system performance very well.

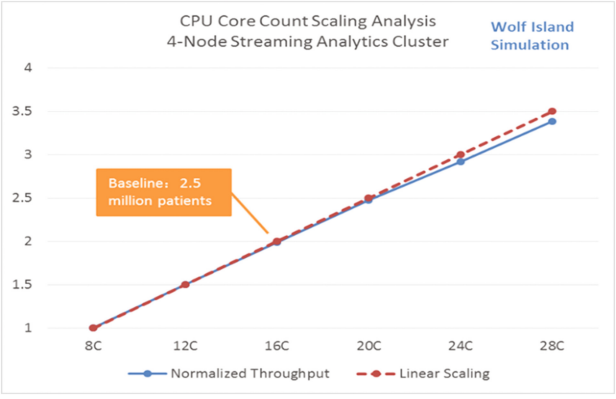


Fig. 12. CPU core count scaling results.

Then we evaluated the system performance with difference CPU type as shown in Fig. 13, where server node with Intel Xeon E5-2687 W V3 (Hasware micro-architecture) could support 0.93 million patients compared to server node with Intel Xeon E5-2650 V2 (IvyBridge micro-architecture) could only support 0.63 million.

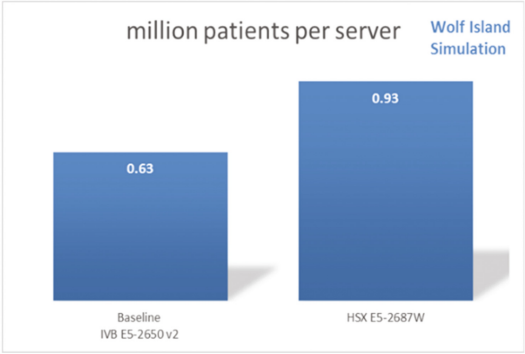


Fig. 13. CPU generation scaling results.

With the limitation of CPU frequency, core count and micro-architecture, cluster performance could only be scaled at a very limited range, if much more patients need to be support, more server node have to be add into cluster like the following scaling.

From the simulation result, to support 10 million patients there should be at least 16 server nodes in the spark cluster as shown in Fig. 14.

In some other cases, cluster performance also need to be scaled down to save cost and power, through decommission some server nodes or slow down the CPU operation frequency. Simulation based performance prediction could also help how to scale the cluster performance down.

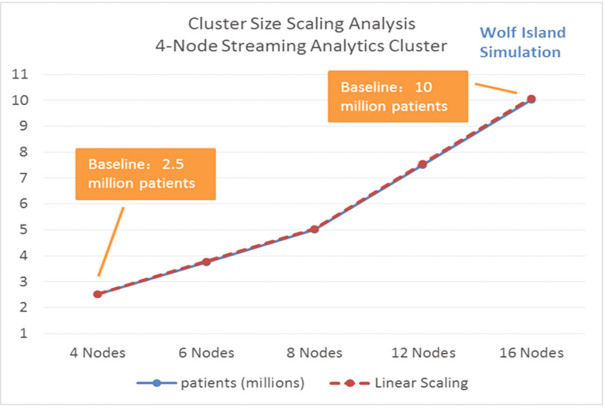


Fig. 14. Node count scaling results.

6 Case Study 2: Memory Optimization for Spark Performance

Memory tuning is critical in Spark. The Spark PageRank optimization is a good candidate to illustrate how memory settings at different layers impact Spark performance, and how simulation based tuning can help optimize Spark application performance. Trade-offs at Spark, and JVM levels are described in this section, and then simulation based optimization solution was shown at the end of this section.

Spark PageRank is memory intensive and generates a large set of intermediate data which pushes up the system memory utilization. These intermediate data are also shuffled across cluster nodes. Shuffle is the operation that moves data point-to-point across machines. It has a critical impact on Spark performance, as shown in the latest Spark core performance optimization work [15]. In the Spark workflow, intermediate data is held in the memory buffer first and then written to disk when the buffer is about to become full (buffer spilling). As the latency of spill data write process is very long, the size of the memory buffer reserved for intermediate data heavily impacts the Spark performance.

The spill buffer is part of the executor JVM heap, whose size is controlled by the Spark parameter shuffle.memory.fraction. If the spill buffer is large enough to hold all the collected data, then no spill occurs, or else, it flushes the buffer first and continue to collect shuffle data. Larger spill buffer size would reduce the number of spill operations hence improving performance. However since the spill buffer space is taken from the JVM heap, a big spill buffer would leave very few memory left for other tasks, such as RDD transfers, that share the same JVM heap space.

Table 5. Cluster hardware and software settings.

Approach	Executor memory	Executor count	Executor VCore
1	6 GB	16	1
2	12 GB	8	2
3	48 GB	2	8
4	96 GB	1	16

Executor memory and VCore which are configured by JVM and YARN settings are also very performance sensitive. We run the simulation with four sets of different memory configurations shown in Table 5 together with 5 different shuffle memory fraction configurations and get following result shown in Fig. 15.

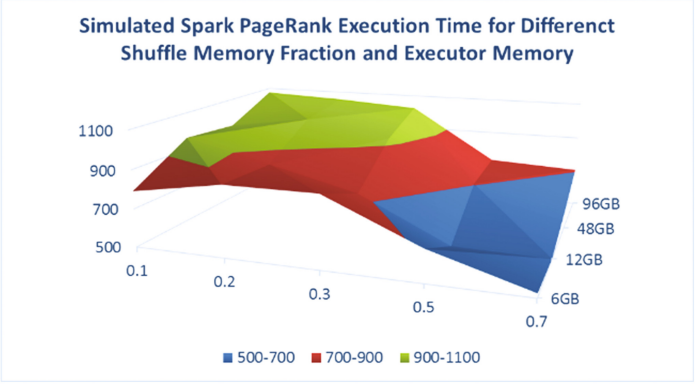


Fig. 15. Simulation based optimization of Spark memory system.

This figure demonstrates how simulation based optimization can be used in a systematic way to explore execution time against the shuffle.memory.fraction and the executor memory size in GB. Best performance is achieved for a 12 GB executor memory size combined with a shuffle.memory.fraction of 0.5. This represent a 71% improvement compared to the default configuration (6 GB, 0.2). We can use the simulator to predict performance for different cluster configuration without real cluster deployment with much higher accuracy than experienced estimation.

There is no general solution that can satisfy all cases but simulation based optimization can be used to thoroughly explore the space of possible solutions so that the best configuration trade-off can be found.

7 Related Work

Several existing simulator are dedicated to simulate the MapReduce computing paradigm, but no Spark simulator is currently available. The most closely related works are based on full system simulators which usually are general purpose functional simulators. One of this kind is Simics-based [16] cluster simulator that can run any kind of unmodified Big Data applications, but these simulators was often used to characterize Spark and other Big Data workloads [17], their simulation speed are very slow especially when the node number of the target cluster increases, what's more Simcs can't provide accurate timing information for cluster applications.

Simflex is based on Flexus simulation engine and SMARTS rigorous sampling engine [18] while Flexus was also built on Simics.

An instruction set simulator-based full system simulator [19] can run unmodified message-passing parallel applications on hundreds of nodes at instruction level, but similarly because it is a low level simulator its simulation performance is poor and it can hardly be used for performance optimization.

Compared to the above mentioned simulators this paper proposes a fast and high accuracy layered simulation framework. Several hundred nodes clusters can even be simulated on a desktop in relative short time.

8 Conclusion and Future Work

As the computing core of next Big Data clusters, Spark plays an important role in capacity planning consideration. It is critical to be able to predict Spark performance accurately and efficiently so that the right design decisions can be taken. This is however a challenging task due to vast hardware diversity and rapidly increasing software complexity. In this paper, we proposed an innovative simulator used to simulate Spark cluster performance at system level.

We have validated its accuracy and efficiency via several widely used benchmarks. Experimental results demonstrate the accuracy and capability of our Spark simulator: the average error rate is below 7% across the scaling of 33 software parameters and 5 group of hardware settings.

The ability to quickly simulate Spark clusters with high accuracy on commodity clients makes our simulator a promising approach as a design tool to perform capacity planning before real deployment. For our 5 nodes 50 GB data set size configuration, simulation times vary between 30 min and 4 h.

Moreover system engineers could also use this simulator to optimize Big Data cluster configuration, maximize cluster performance, evaluate server design trade-offs and make system-level design decisions.

For easier Spark development, the Spark ecosystem brings additional functionality like MLib (machine learning library), GraphX, Spark Streaming and Spark SQL. We will extend our Spark model to these functionalities. As heterogeneous architecture are more and more popular in data center these days, we will also extend our capability to support the simulation of heterogeneous systems like XEON + FPGA based clusters.

References

1. http://spark-summit.org/wp-content/uploads/2014/07/Sparks-Role-in-the-Big-Data-Ecosystem-Matei-Zaharia1.pdf
2. https://spark.apache.org/
3. Bian, Z., Wang, K., Wang, Z., Munce, G., Cremer, I., Zhou, W., Chen, Q., Xu, G.: Simulating big data clusters for system planning, evaluation and optimization. In: ICPP-2014, 9–12 September 2014, Minneapolis, MN, USA (2014)
4. Xin, R.S., Rosen, J., Zaharia, M., Franklin, M.J., Shenker, S., Stoica, I.: Shark: SQL and rich analytics at scale. In: SIGMOD (2013)
5. Zaharia, M.: Spark: in-memory cluster computing for iterative and interactive applications. In: Invited Talk at NIPS 2011 Big Learning Workshop: Algorithms, Systems, and Tools for Learning at Scale (2011)
6. Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I.: Spark: cluster computing with working sets. In: HotCloud 2010 Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing, p. 10. CA (2010)
7. Apache Software Foundation: The Apache Software Foundation Announces Apache Spark as a Top-Level Project, 27 February 2014. Accessed 4 Mar 2014
8. Kolberg, W., Marcos, P.D.B., Anjos, J.C., Miyazaki, A.K., Geyer, C.R., Arantes, L.B.: MRSG – a MapReduce simulator over SimGrid. Parallel Comput. **39**(4–5), 233–244 (2013)
9. Wang, G., Butt, A.R., Pandey, P., Gupta, K.: A simulation approach to evaluating design decisions in MapReduce setups. In: Proceedings of the 17th Annual Meeting of the IEEE/ACM International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2011), London (2011)
10. Kennedy, P.R., Gopal, T.V.: A MR simulator in facilitating cloud computing. Int. J. Comput. Appl. **72**(5), 43–49 (2013). Published by Foundation of Computer Science, New York, USA
11. Verma, A., Cherkasova, L., Campbell, R.H.: Play It Again, SimMR! In: Proceedings of IEEE International Conference Cluster Computing (Cluster 2011) (2011)
12. Intel, Simulation software. http://www.intel.com/content/www/ru/ru/cofluent/intel-cofluentstudio.html
13. Skiena, S.S.: The Algorithm Design Manual, Springer (2008)
14. https://www.phdata.io/real-time-analytics-on-medical-device-data/
15. https://databricks.com/blog/2015/04/24/recent-performance-improvements-in-apache-spark-sql-python-dataframes-and-more.html
16. Magnusson, P.S., Christensson, M., Eskilson, J., Forsgren, D., Hallberg, G., Hogberg, J., Larsson, F., Moestedt, A., Werner, B.: Simics: a full system simulation platform. IEEE Comput. **35**(2), 50–58 (2002)
17. http://prof.ict.ac.cn/BigDataBench/simulatorversion/
18. http://parsa.epfl.ch/simflex/overview.html
19. León, E.A., Riesen, R., Bridges, P.G., Maccabe, A.B.: Instruction-level simulation of a cluster at scale. In: HPCC, 14–20 November 2009, Portland, OR, USA (2009)

Generator Platform of Benchmark Time-Lapsed Images Development of Cell Tracking Algorithms: Implementation of New Features Towards a Realistic Simulation of the Cell Spatial and Temporal Organization

Leonardo Martins[1(✉)], Pedro Canelas[1], André Mora[1], Andre S. Ribeiro[2], and José Fonseca[1]

[1] Computational Intelligence Group of CTS/UNINOVA, Faculdade de Ciências e Tecnologia da, Universidade Nova de Lisboa, Quinta da Torre, 2829-516 Caparica, Portugal
{l.martins, p.canelas}@campus.fct.unl.pt, {atm, jmf}@uninova.pt

[2] Laboratory of Biosystem Dynamics, Department of Signal Processing, Tampere University of Technology, Tampere, Finland
andre.ribeiro@tut.fi

Abstract. Recent developments in live-cell microscopy imaging have led to the emergence of Single Cell Biology. This field has also been supported by the development of cell segmentation and tracking algorithms for data extraction. The validation of these algorithms requires benchmark databases, with manually labeled or artificially generated images, so that the ground truth is known. To generate realistic artificial images, we have developed a simulation platform capable of generating biologically inspired objects with various shapes and size, which are able to grow, divide, move and form specific clusters. Using this platform, we compared four tracking algorithms: Simple Nearest-Neighbor (NN), NN with Morphology (NNm) and two DBSCAN-based methodologies. We show that Simple NN performs well on objects with small velocities, while the others perform better for higher velocities and when objects form clusters. This platform for benchmark images generation and image analysis algorithms testing is openly available at (http://griduni.uninova.pt/Clustergen/ClusterGen_v1.0.zip).

Keywords: Microscopy · Synthetic time-lapse image simulation · Cell tracking · Cluster tracking

1 Introduction

Recent advances in live-cell microscopy imaging have enhanced images quality and triggered the development of techniques for detecting and observing cellular structures and their kinetics [1, 2].

M.S. Obaidat et al. (eds.), *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing 676,
https://doi.org/10.1007/978-3-319-69832-8_4

Live-cell imaging entails the tasks before and during image acquisition at the microscope, which includes image refinement, such as tuning illumination, focus, drift correction, stage positioning and microscope components selection (e.g. shutter, lens, camera, stage) [3]. It follows image processing (e.g., registration, segmentation, tracking, statistical quantification and background correction) [3, 4].

Microscopy software packages include automatic correction algorithms for noise attenuation, contrast correction, illumination compensation, etc. [5].

The first step is usually image registration (overlay of two or more images of the same object at different instants, viewpoints or sensors) is a classical step with several methods available, which are based on modality, intensity, type of data, dimensionality, domain and type of transformation, and registration methodologies [6, 7].

The next step is the segmentation of cells or cellular structures of interest [8], where these segmented objects are detected, located, and separated from the background. The main challenge here is to automatize it with high specificity and sensitivity for a wide number of cases. Presently, there are various approaches, such as intensity thresholding, feature detection, morphological filtering, region accumulation, deformable model fitting, etc. [8].

When handling a time series, one needs track the objects between frames, i.e., to link the segmented objects in the actual frame with the ones from the previous frame, so as to attain the object's trajectory. With the information describing the target defined by the state sequence X_k, $k \in \mathbb{N}$ (where $\mathbb{N}$ is the set of frames), and the measurements defined by Z_k, the goal of tracking is to estimate X_k, given all measurements until the moment $Z_{1:k}$ [9]. This is made difficult by noise, occlusions, illumination changes, complex motions and object's shape dynamics, which can enhance the misidentification of object tracks [10].

Currently available tools for tracking in different microscopy settings include the 'Cell-C', based on DAPI staining and fluorescence *in situ* hybridization images [11], 'CellTracer', which applies morphological methods to automatically segment bacterial cells, yeast and human cells [12], 'MicrobeTracker' and its accessory tool 'SpotFinder', which segment *Escherichia coli* and *Caulobacter crescentus* cells and detected fluorescent spots within [13], 'Schnitzcells', which segments and tracks *E. coli* cells in confocal or phase contrast images [14] and, 'CellAging', which was developed for cell segmentation and tracking in order to study the segregation and partitioning in cell division of protein aggregates [15].

The validation of these tools requires gold-standard images, usually manually annotated by biology experts. However, this validation is problematic, as it is expert-dependent (both inter-user and intra-user variability can be high) and is impractical in high-throughput data-sets [16]. To overcome this problem, a viable alternative is using artificial images of biologically inspired objects. These images, whose ground truth is known, can be used for the accurate quantitative evaluation of the image processing algorithms [4].

Next, we provide a comprehensive literature review of existing tools for simulation of synthetic microscopy images and of recent developments on cell tracking algorithms. In Sect. 3 we present the contributions to the development of the image simulation tools (models and parameters) and the implementation of three different tracking algorithms. In Sect. 4, the tracking results of several examples are presented

using different parameters. Finally, in Sect. 5, we present our final remarks on the development of simulation tool and the results of the three algorithms, along with a description of potential future endeavors.

2 State of the Art

2.1 Synthetic Image Generators

There have been several contests and open challenges on microscopy image processing, usually requiring that each methodology is tested on the same benchmark data-sets (acquired by an independent laboratory or created by artificial image generators) [8]. Such artificial image generators require realistic biological models, and commonly use theoretical and experimental information on the statistical distributions of the object's behavior [17] and spatial and temporal data [18, 19]. If the object studied is a cell, these models should include morphology parameters such as cell shape and size, location of subcellular structures, kinetic and spatial statistics of cell growth, cell division, cell migration and models of internal cell functions.

The architecture of microscopy image simulators based on biological models can be divided in three main stages: the digital phantom object generation, the simulation of the signal passing through the optical system and the simulation of the image formed on a specific sensor [20].

Simulators such as 'SIMCEP' [21] have provided a gold-standard platform to validate and test image processing tools, such as the previously mentioned 'CellC' [11], the open-source and Java-based image processor ImageJ, and the commercially available MCID Analysis (Imaging Research Inc., Catharines, ON, Canada; Evaluation ver. 7.0), along with other image processing tools [22]. The phantom objects are generated with different cell parameters, such as probability of clustering, cell radius, and cell shape and with parameters related to the sensors and the optical system, such as background noise and illumination disturbance [22, 23].

'CytoPacq' is another toolbox specifically developed to simulate all three phases. For that, it is equipped with three different modules. The first module ('3D-cytogen') generates the digital object phantom, which imitates the cell structure and behavior generating microspheres, granulocytes, HL-60 Nucleus and images of Colon Tissue. The second module ('3D-optigen') simulates the transmission of the signal through the lenses, objective, excitation filter and emission filter (various sets of equipment can be simulated). The last module, '3D-acquigen' is the digital CCD camera simulator of the image capture process (noise, sampling, digitization) by changing the camera selection, the acquisition time, the dynamic range usage and the stage z-step [20, 24]. The same group also introduced a novel versatile tool ('TRAgen'), capable of generating 2D time-lapses by simulating live cell populations as a ground-truth for the evaluation of cell tracking algorithms. In this work, they included models of cell motility, division and clustering up to tissue-level density [25]. Both simulators have been an important step in the simulation of cellular dynamics, such as intracellular protein or RNA levels or even cell migration, division and growth [2, 3].

Another toolbox, called 'SimuCell' [26], is capable of generating artificial microscopy images with heterogeneous cellular populations and diverse cell phenotypes. Each cell and their organelles are modeled with different shapes and distinct distributions of biomarkers over each shape, which can be affected by the cell's microenvironment, demonstrating the importance of good cell placement (e.g. in clusters, overlapping existing cells) [26].

The 'CellOrganizer' toolbox was developed based on laboratory data and using machine-learning techniques to generate the entire cell, including structures such as the nucleus, proteins, cell membrane and cytoplasm components [27]. Although the learn-based model was capable of extracting a very precise shape model, it cannot be described in precise mathematical terms [28].

Most image generators have focused on the simulation of morphological features and spatial information of the cell. Morphological information can suffice to create multidimensional images, but it cannot simulate time-lapsed multimodal and functional images, where important time-dependent processes are present. To simulate such images of bacterial cells, the 'miSimBa' (Microscopy Image Simulator of Bacterial Cells) tool has been under development [29]. The simulated images can reproduce spatial and temporal bacterial time-dependent processes by modeling cell growth, division, motility and morphology: shape, size and spatial arrangement [29]. Relevantly, these simulation tools can also be used to generate "null-models" [30], to study statistical patterns in absence of a particular mechanism (e.g. removing the nucleoid to study how it influences the spatial distribution of protein aggregates).

2.2 Cell Tracking

Several tracking methods have been proposed, differing on how they process available object features, type and number of tracked objects [10]. In order to decide which approach to follow, the object's representation, defined during the segmentation process, must be taken into account. Objects can be represented through points, geometric shapes, silhouette and contour, articulated shape model or skeletal model, leading to different developmental approaches [10]. Tracking methodologies were divided into three main categories: Point Tracking, Kernel Tracking and Silhouette Tracking [10].

Objects in Point Tracking are represented by points and tracked based on their position and motion. The main issues of this methodology are the presence of occlusions and the entries and exits of objects in the field of view. This category has been divided in Deterministic and Statistical methods. Deterministic methods associate each object with the application of motion constraints, while statistical methods take into account random perturbations and noise during the tracking process [10]. The Nearest-Neighbor (NN) algorithm is the source of all deterministic approaches and uses only the distances between objects in k and k−1, matching the objects with the smallest distances. This distance can be based on position, shape, color and size [31].

An efficient visual object tracking algorithm was proposed by [32] that combines NN classification with descriptors based on the scale-invariant feature transform, efficient sub-window search and an updating and pruning method to achieve balance between stability and plasticity. This method successfully handles occlusions, clutter, and changes in scale and appearance.

The probabilistic data association filter (PDAF) and the joint probabilistic data association filter (JPDAF) are the basis for the statistical methods. PDAF uses a weighted average of the measurements as input, modeling only one target and considering linear dynamics and measurement models. JPDAF is an extension of PDAF, allowing multiple target tracking. The assumptions are the same when calculating the target's association probabilities jointly. In both methods, if the model is linear, then the Kalman Filter has a relevant influence. One of the problems of these methods is the incapacity to recover from errors, because only the last measurement is used [31]. The Kalman filter is an optimal estimator, which means that it assumes parameters from indirect, inaccurate and uncertain observations and if all noise is Gaussian, the linear Kalman filter minimizes the mean square error of the estimated parameter. This filter is widely used to obtain the optimal state estimate [31].

A different method [33] combining the JDPAF and a particle filtering [34] was proposed and was named 'Monte Carlo JPDAF'. This method uses three models: the first with near constant velocity, the second with near constant acceleration and a third with both models, which achieved the best performance.

Another statistical method is the multiple hypothesis tracking (MHT), which is one of the most used with point features, but has computational limitations both in time and memory [9]. This method postpones data association until enough information is available. The MHT starts by formulating all possible hypotheses, which develop into a set of new hypotheses each time new data arrives, generating a tree of hypothesis [31]. For each hypothesis, the position of the object in the next frame is predicted and then compared with the measurements, calculating their distance. The associations are made for each hypothesis, generating new hypotheses for the next iteration [10]. The tree of hypotheses should be cut, because it grows exponentially with the measured data. This can be done by clustering, i.e., measurements are subdivided into independent clusters. If a measurement cannot be associated with an existent cluster, a new one is created. Another way of cutting the tree is pruning, meaning that as new iterations are added, a part of the tree is deleted [31].

Unlike PDAF and JPDAF, the MHT method can deal with objects entering, exiting and being occluded from the field of view. Kernel Tracking can be done using templates and density-based appearance models or multi-view appearance models. Templates use basic geometric shapes, while multi-view models encode different views of the object. Mean shift and KLT (Kenade-Lucas-Tomasi) are examples of template and density-based appearance models [10].

In mean shift, the appearance of the objects being tracked is defined by histograms. Similarities are measured using the Bhattacharyya coefficient [35] and the Kullback-Leibler divergence [36]. The process tries to increase similarity between histograms, by repeating each iteration until they converge [37].

KLT is an optical-flow method, which uses vectors to show the changes in the image (i.e. translation). A version of this method was proposed in which the translation of a region centered on an interest point is iteratively computed. Then, the tracker evaluates the tracked patch, computing a transformation in consecutive frames [38]. These methods are effective while tracking single objects, but have problems dealing with multiple objects. Silhouette Tracking consists in using precise information about the shape of the objects, using Shape Matching and searching for an object silhouette

and its model in each frame. Each translation from frame to frame is handled separately by finding corresponding silhouettes detected in two consecutive frames. Another approach is based on the evolution of the object contour, connecting the correspondent objects by state space models or by minimizing the contour energy [10].

When tracking objects, one usually obtains multiple measurements. The incorrect ones are referred to as false measurements or clutter. The measurement with highest probability of being originated from the tracked object is then selected. If the algorithm selects the wrong measurement or if the correct measurement is not detected, a poor state is estimated. To solve this issue (reducing the computational cost), a validation region (measurement gate) is selected. The measurement gate is a region in which the next measurement has a higher emergence probability [31].

3 Methodologies

3.1 Implementation of the Image Generator - Tool Interface and Basic Functionalities

The image generator interface and the tracking methods were implemented using the C# language from Visual Studio 2015. This sub-section focuses on the implementation of the image generator and its basic features. In order to facilitate the analysis of the tracking algorithms an intuitive interface was designed. The time-series generator allows the user to change a number of settings such as the number of objects, frames, clusters, and their features. The generator automatically creates a csv file containing the object's properties (position in x and y coordinates and a shape-related factor called "morphology", which is a rational number between 0 and 1 as defined in the Object Shape Sub-Section). The tool interface is shown in Fig. 1. At the top row of the

Fig. 1. Image generator tool interface.

window there are frame handlers, to advance forward and backward in the time-series, or to go directly to a specific frame. The "Time-Lapse" button reproduces the full time-series with a frame-rate of 25 frames/second.

The left bar contains the boxes to write the desired width and height of images, in pixels. The user can also choose the number of objects in each frame, and the total number of frames. The "Maximum Velocity" is the maximum distance, in pixels, that an object can travel between frames, while the "Maximum Morphology Difference" is the maximum difference of the "morphology" factor that an object can have between frames, in percentage. The "Physical Move" button controls the option of giving objects physical limitations to their kinetics. If it is selected, each object has a velocity and orientation assigned to it, meaning that its position dynamics will depend on these two variables. If it is not selected, objects will move arbitrarily between frames.

One can also select "Allow Entries/Exits", to allow the objects to enter and exit the image limits. If unselected, objects collide and are reflected by the edges of the image when reaching them. When the option "Allow Occlusions" is selected, objects move without restrictions due to superposition between them. If it is not selected, objects collide between them similarly as when colliding with the edges.

Objects clustering can also be forced checking the "Create Clusters" option. When selected, all objects of each cluster have the same physical features. In this setting, "Physical Move" is automatically selected and "Allow Occlusions" is deselected, blocking the correspondent checkboxes. The button "Cluster Properties" (shown in Fig. 5) leads to a new window with the options for clusters' creation. Here, the desired number of clusters, objects per cluster, and size of the clusters in pixels can be selected. It is also possible to choose between two types of objects' kinetics: "Follow the Leader" and "Alternative Movement". The application of "Cluster Centre Force" and its strength are shown in Fig. 6 and explained in the Sub-Section Cluster Creation.

3.2 Object Modeling

This sub-section focuses on the modeled features, namely object shape, movement, growth, division and clustering, which were improved from the previous toolbox towards a realistic simulation of the bacterial cell spatial and temporal organization.

Object Shape

To create a realistic simulation of bacterial cells, we first need to investigate how they are classified by their shape. Bacterial cells can have a spherical shape (coccus) a rod-shape (bacillus), while other bacteria have shown a vast diversity of shapes, such intermediate shapes (coccobacillus) or curved/corkscrew shapes (spirochete, spirillum and vibrio), or even square and star shapes, each of them with its specific purpose [39, 40].

Bacteria can also have a wide range of cell sizes (volumes that range from 0.02 to 400 μm^3), where even a vast variability can be observed within the same species [41, 42]. These variations can be explained due to cell adaptation to external factors, such as lack of nutrients leading to starvation, situations of extreme temperatures (low and high) or of extreme dryness [42].

A typical bacterial cell envelope is mainly composed by a cytoplasmic membrane and peptidoglycan (also known as murein) cell wall. Bacteria can also be divided in two groups regarding a fundamental difference in the cell envelope: Gram-negative and Gram-positive bacteria. In the first group (which is the case of *E. coli*) a bacterial outer membrane is also present (with intercalating pore-forming proteins, called porins), with lipopolysaccharides connected to the exterior of that outer wall. The interior of the outer wall is then connected to a very thin murein wall by a lipoprotein [43]. In the second group (which is the case of human pathogenic bacterium Streptococcus pneumonia), the cell envelope consists of a very thick murein wall (sometimes more than 10 times thicker than the first group) with teichoic acids spread across the murein. The shape is maintained and determined by the way murein is incorporated during cellular elongation, especially in rod-shape organisms, such as *E. coli* [44] and *B. subtilis* [45], as the murein is the main cell wall structure that supports the stress from the outside [46], as computational physical models have been develop to study how defects in the murein can affect *E. coli* shape (and the shape robustness to murein damage) and how different murein defect patterns can build bacterial shape patterns such as curved rods and spirochaetes [47]. Along with the cell wall, other cytoskeleton proteins are associated with bacterial shape, such as FtsZ (tubulin homologue), MreB (actin homologue) and crescentin [39, 45] (Fig. 2).

Fig. 2. Example of bacterial cell shapes. Spherical shape (coccus) in dark gray, a rod-shape (bacillus) in orange, intermediate shape (coccobacillus) in green and curved shapes (spirochete, spirillum and vibrio) in blue.

In the first version of this generator [48], objects were just represented by circles and the morphology factor (radius), which only represented coccus shaped bacteria. There was a conversion factor that determines the maximum radius of the objects (corresponding to morphology value 1). By default, this factor was initialized at 30. The development towards realistic bacterial cells will involve the representation of objects with this variables: *Object_ID* (identifies each object), *MajorAxisSize* (size of Major Axis), *Orientation* (defines the orientation of the Major axis), MinorAxisSize (size of Minor Axis), *Curvature* (0 if you want to create line objects and 1 if both ends of the Major Axis touch, transforming the long bacteria into a circle) *IDPixelList* (this variable populates all pixels that correspond to the object), *Centre* (center of mass of the object), *Division* (this value starts at 0, changes to the time step where the division event occurred), *Parent* (equal to its *Object_ID* in every time-step except in the time-step after a division event, which is equal to the parents *Object_ID*). With these

new parameters we are able to change the shape of the cell towards more realistically bacterial shapes. Cells with similar *MajorAxisSize* as *MinorAxisSize* will have coccus shape, when we increase the *MajorAxisSize*, we will get intermediate shapes (coccobacillus) and with large increases of *MajorAxisSize* we will have bacillus shapes. Using the *Curvature* and large *MajorAxisSize* we will create curved shaped objects. These properties are populated for each time-step of the simulation and can be changed by events such as cell growth, division, motility and clustering, as explained in the next sub-sections.

Object Growth

Bacterial cell cycle is normally divided in three stages, specifically a period between its "birth" and the initiation of DNA replication, a replication period when the cell increases its mass and size (Cell Growth) and, finally, a binary fission process into two new daughter cells (Cell Division), which is repeated over the next generations [49].

The creation of new murein polymer can lead to cell growth through cell elongation, as murein is inserted in the sidewalls at the middle of the cell or at the poles. The creation of the division septum at the mid-cell then leads to a division event (this is also the main process for cell growth in spherical cells, where cell elongation does not occur), where two daughter cells are created [39]. Each of those processes have their own protein and enzymatic apparatus, working in specific places of the cell wall [39, 45]. The FtsZ cytoskeleton protein along with various other proteins create the division septum at the middle of the cell (as two proteins MinC and SlmA that are present in the rest of the cell, inhibit the assembly of the FtsZ ring required for division [50].

In the first version of this generator [48], the morphology shape-related factor called was set at 0.05 (this value was chosen to emulate biologically inspired objects that slowly change their shape over time). Although this process emulates how other cell shapes (bacillus, coccobacillus, vibrio) change their cell size, this actually needs to be changed in truly spherical shaped cells (cocci) as they do not have an elongation process [51], but create a division septum at mid-cell, which allows them to create two daughter cells roughly of the same size of the parent cell due to entropic forces [39]. For the remaining shapes, we implement the creation of new pixels along the Major Axis as the growth process.

Object Division

In the first version of this generator [48], no division process was implemented. This new version has implemented object division This feature is intended to be an approximation to living cell proliferation, where a parent cell "splits" in half, originating two daughter cells. In this specific case, since objects are represented only by circles, not by complex shapes, division consists in splitting an object with a morphology factor m into two objects with a factor m/2.

There was a factor named "Division Probability", measured in percentile that defines the probability of occurring a division for each object, in each frame of the time-series, which happens stochastically. The daughter objects inherit from the parent the physical parameters share the same cluster force (if inside a cluster). An example of an object division is shown in Fig. 3.

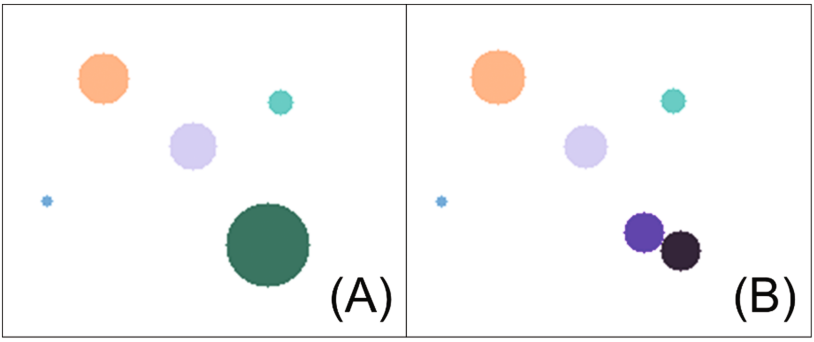


Fig. 3. Example of object division from frame (A) to frame (B).

Object Motility

Bacterial growth as a colony can also be dependent on the capability to move in the direction of more favorable conditions, which at its basic form is normally associated with Brownian random movement or active movement towards a specific gradient, e.g. chemicals (chemotaxis) and temperature (thermotaxis) [52].

According to the user's selection, objects can have movement respecting a number of physical rules. If this option is deactivated, objects will move arbitrarily through the image. In each frame, each object can move to a new x and y coordinates by an arbitrary distance that cannot be higher than the "Maximum Velocity" value in pixels.

If entries and exits are deactivated and if an object is heading to the image boundary, it is reflected respecting Snell's Law, as seen in Fig. 3 causing a change in the angle's direction of movement. If occlusions are deactivated, when two objects are about to collide, they change to opposite orientations in an approximation to the reflection laws, but ignoring differences in their morphologies (Fig. 4).

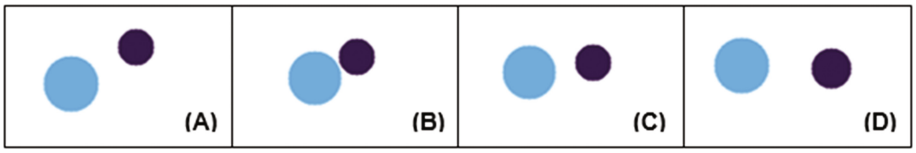


Fig. 4. Collision between objects with "Physical Move". Objects in: (A) Frame 10; (B) Frame 16; (C) Frame 19; (D) Frame 23.

With occlusions and "exits and entries" also deactivated, objects will avoid the positions where they collide with other objects or go out the image boundaries, searching for a position considering these limitations and the maximum distance they can move between frames. If the user chooses to give objects "Physical Move", in addition to previous features, each object will have a velocity and an orientation assigned to it, meaning that their position dynamics will depend on these two values. In each frame, each object will have new x and y coordinates distanced "d" (no bigger

than "Maximum Velocity") from the previous frame coordinates, direction "o" (between 0 and 2pi radians), with both components using an independent random variable, consistent with the Brownian random movement. Collisions between objects might need to be reconsidered as bacterial cells tend to create clusters when they bump with other cells, and not move away from those cells.

Cluster Creation

In terms of spatial arrangement, bacteria can be organized in single forms or be grouped in pairs (diplo prefix), in chains (strepto prefix). Cocci bacteria can also organize in groups of 4 (tetrad), 8, 16 or 32 (sarcinae) or in grape-like clusters (staphylo prefix). Bacilli bacteria can organize in palisade structures (side by side) or can be in unstructured spatial clusters [40].

When selecting the option "Create Clusters", the Generator will create a time-series with the number of clusters, objects and size of cluster chosen by the user. These options (shown in Fig. 5) must be consistent and take into consideration the image size.

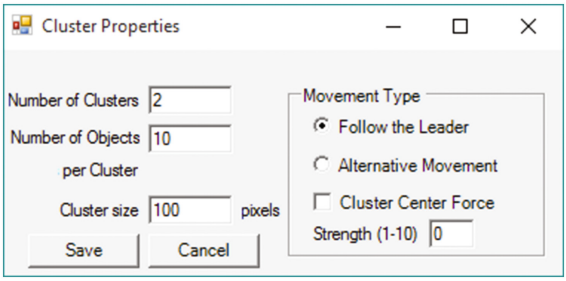


Fig. 5. Interface options for cluster properties.

In "Alternative Movement" (as shown in Fig. 6A) all objects of each cluster have the same physical parameters, which means that they move in the same direction with the same speed (with a small independent arbitrary component).

In the "Follow the Leader" movement mode (as shown in Fig. 6B), each cluster has a leading object. The characteristics of the other objects of the same cluster are dependent on the leader's behavior. The leader "receives" the physical parameters at first frame (velocity and orientation) and at each frame the other objects of its cluster will move in the leader's direction, minimizing the distance to it, but respecting the "non-collision" rule. If two objects from different clusters collide, one of them will start belonging to the other cluster. This may cause the "merging" of clusters.

The "Cluster Centre Force" feature is exclusively for "Alternative Movement" that creates an attraction force at the cluster's center, with a selectable strength selected by the user. This force keeps cluster's objects together, even when colliding with the image borders or other objects. Increasing the strength, the objects will move faster to the cluster's center. In this mode of motility, when objects from different clusters collide, they will be "left behind" by their cluster until they can join it again.

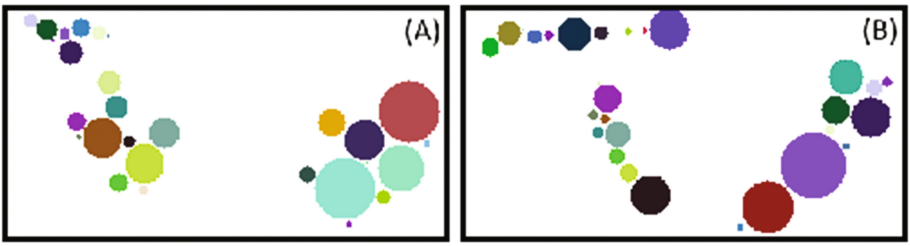


Fig. 6. Exemplificative frames of (a) 'Alternative' Movement (b) 'Follow the leader' Movement.

3.3 Tested Tracking Algorithms

In this Section, we give a small introduction to Nearest-Neighbor Algorithms that were used to test our image generation tool, namely the Simple Nearest-Neighbor (NN) and the Nearest-Neighbor with Morphology (NNm) Algorithms. We also introduce the Density-Based Spatial Clustering of Applications with Noise (DBSCAN), which is mainly used to track clustered objects.

Simple Nearest-Neighbor Algorithm
The first tracking algorithm tested was the Simple NN. This method only takes into consideration the position of each object in each frame of the time-series, and uses the Euclidian Distance between points to find matching objects between frame n and n + 1. Being d_p the distance between two objects:

$$d_p = \sqrt{(x_n - x_{n+1})^2 + (y_n - y_{n+1})^2} \tag{1}$$

Where x_n and y_n are the positions of each object in frame n and x_{n+1} and y_{n+1} are the positions in frame $n + 1$. Having the distance between each object in frame n and all objects in frame $n + 1$, correspondences are made based on the minimum distance. The object in frame $n + 1$ closer to each object in frame n is assigned to it. If two objects in $n + 1$ are assigned to the same object in n, the closer object is assigned, until all correspondences between frames are unique [31].

Nearest-Neighbor with Morphology Algorithm
The NNm algorithm accounts not only for the differences between the positions of each object in each frame, but also for a shape-related factor, called morphology. This algorithm calculates the distance percolated by each object between frames n and $n + 1$ using Eq. (1). Being m_n the morphology of each object in frame n, and m_{n+1} the shape factor in $n + 1$, the difference, d_m, between these variables is calculated by:

$$d_m = |m_n - m_{n+1}| \tag{2}$$

The total difference, d_t, between each object in each frame pair is given by (3) with α and β being the weights given to each partial distance. Here different weights are used (as presented in the Results section), in order to study the best way to combine them:

$$d_t = \alpha \cdot d_p + \beta \cdot d_m \tag{3}$$

Cluster Tracking

Identifying clusters is one of the most complex issues of image characterization [53]. In this work, the problem lays in tracking objects knowing that they are grouped in clusters. Since bacteria often group this way, the goal is to find a method that improves tracking of clustered objects. One of the main problems of clustered objects is illustrated in Fig. 7A. Using NN (or NNm) to track these frames, the algorithm will immediately misidentify at least two of the objects of frame $n + 1$. This will occur in objects $1'$ and $3'$, and it happens because their position in $n + 1$ is exactly the same that objects 2 and 4 have in n.

To solve this problem we implemented a tracking algorithm that considers the cluster's features and its singularities. The first step of this method to track clustered objects is to correctly identify the clusters in the image and the objects belonging to each of them. The adopted method was the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [55] in its revised version [54]. This method formalizes the notion of "cluster" and "noise", using the definition of density to characterize clusters, meaning that to define a cluster, the density of the neighborhood of each point has to be higher than a given threshold. '*MinPts*' is the minimal number of objects in the neighborhood, and Eps is the neighborhood radius (see Fig. 7B).

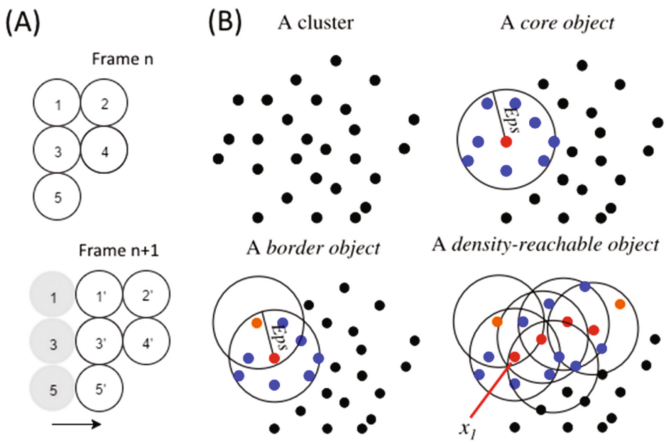


Fig. 7. (A) Example of a possible misidentification using the NN Algorithms. (B) 'MinPts' is defined as the minimal number of neighborhood objects, and Eps as the neighborhood radius, a core object (Red) is defined when its local density is higher than 'MinPts' and a border object (Orange) in its local density is less than 'MinPts'. Two density-reachable objects are defined if a chain of core objects exists with distances between them smaller than Eps. Adapted from [54].

Objects can be divided into three categories: core, border and noise (see Fig. 7B). An object is a core object if its local density is higher than '*MinPts*'. It is considered a border object if its local density is less than '*MinPts*' and it belongs to the neighborhood of a core object. An object is classified as noise if in its Eps radius there are less than '*MinPts*' objects and none is a core. Finally, we identify two density-reachable objects if there exists a chain of core objects between them (see Fig. 7B), with distances between them smaller than '*MinPts*' [54].

This approach improves clustering identification when the data has dense adjacent clusters [54]. They also introduced the concept of core-density-reachable objects, which is similar to the chain of density-reachable objects, but cutting border objects from chain's ends and staying unclassified until all core objects are identified [54].

The algorithm has two main steps: '*dbscan*' and '*ExpandCluster*'. The first step lies in covering each object and running '*ExpandCluster*' if the object is unclassified. Then, it returns all objects that are core-density-reachable from that one. If it is a core object, a cluster is produced. If it is a border object, it has no core-density-reachable objects, and proceeds to the next one. After all chains from the core object are known, it is assigned to its best density-reachable chain and all border objects.

After identifying the clusters in all frames with DBSCAN, a novel algorithm for object tracking was developed. This algorithm assumes that objects are grouped and move in clusters, treating each cluster as a separate individual while tracking. The first step (with all clusters identified) is to isolate the clusters and calculate their centroid, in coordinates x and y:

$$x_{centroid} = \sum_{i=1}^{N} x_i / N \tag{4}$$

After all centroids are calculated, they are processed as objects, since they have their own coordinates. The NN algorithm is then applied to these coordinates, tracking each cluster individually and resulting in a sequence of results similar to object tracking.

4 Results and Discussion

We generated several time-series that can be used as a benchmark to test tracking algorithms. For this, we simulated examples with different starting number of objects (20 to 160) and 'Maximum Velocity' (V = 5, 10, 15, 20 and 30).

The generated images have a 1000 × 500 pixel size (first and second experiment) and 1500 × 100 (third experiment). The implemented Tracking Algorithms automatically processes the csv files with the objects' true positions produced by the Image Generator. The detected object tracking is then compared with the gold standard. In this comparison, a False Positive (FP) is counted when one object is incorrectly tracked from one frame to another and a True Positive (TP) is accounted when one object is tracked correctly between two consecutive frames. It is important to notice that errors that occur in the beginning of the time series are typically propagated through the entire sequence. We present in the following tables the tracking error (false discovery rate), calculated as FP/(FP + TP).

4.1 Simple Nearest-Neighbor Algorithm

We tested 10 time-series of 100 frames for each example with different objects and different maximum velocity. In Table 1, we present the tracking performance of the Simple NN algorithm, based on the ground-truth produced by the image generator. The tracking error is calculated on every frame and accumulated until the end of the time-series. In this case, the morphology shape-related factor called was set to 0.05 (this value was chosen to emulate biologically inspired objects that slowly change their shape over time). The results from Table 1 show that this simple algorithm can handle the increase in the number of objects while keeping a small velocity, and that when raising the velocity to 20 and 30 the tracking performance was significantly reduced.

Table 1. Tracking errors of the Simple Nearest-Neighbor Algorithm.

Obj.	V = 5	V = 10	V = 15	V = 20	V = 30
20	0,00	0,92	1,06	4,19	19,20
40	0,26	1,27	3,23	5,93	24,01
60	0,06	1,58	5,63	12,38	39,66
80	0,24	1,84	6,62	15,74	45,06
100	0,27	1,20	7,85	19,94	49,76
120	0,22	1,69	10,57	21,16	51,86
140	0,55	3,71	14,16	26,57	58,07
160	0,42	4,12	14,91	33,74	63,89

4.2 Nearest-Neighbor with Morphology Algorithm

In this second experiment, we show how tracking taking into account the morphology of the object can be helpful in the worst case scenario of the last experiment. In Table 2, we present the results of the tracking performance of the NNm Algorithm. In this case we also produced 10 time-series of 100 frames for each example with different objects and different maximum velocity, but also with distinct morphology factors.

We tested the algorithm in two configurations; the first giving a 60% importance to the calculated distance between objects (α factor in Eq. 3) and 40% to the calculated morphology difference (β factor). For the second configuration we used 40% for α and 60% for β. The impact of the shape-related factor was also studied using both 0.05 and 0.1. For this section we extended our results [48] to include lower velocities and less objects when comparing our analysis against the simple NN algorithm. From Table 2, we observe that tracking results are improved by using the NNm Algorithm (e.g. in the worst case scenario the error percentage was reduced from 64% to 47%) for the m factor = 0.05 case, but at lower velocities, the Simple NN algorithm achieves similar results (compared with NNm) even with a large number of objects.

It is important to note that, as most of bacterial cells in live-cells imaging are placed in agarose gel, where they do not move very fast, but they are able to grow and create large clusters of cells, in cells that have large movement capabilities, other tracking algorithms need to be used and compared. We also should note that the second configuration (40% for α and 60% for β) gave better results than the first one, so giving

Table 2. Tracking errors of the Nearest-Neighbor with Morphology Algorithm.

Obj.	m factor = 0.05					m factor = 0.1				
	V = 5	V = 10	V = 15	V = 20	V = 30	V = 5	V = 10	V = 15	V = 20	V = 30
α = 60% *and* β = 40%										
20	0.00	0.92	0.00	2.27	11.28	0.00	0.00	1.43	0.37	14.17
40	0.00	0.46	2.45	3.36	18.10	0.00	0.33	2.00	8.71	21.06
60	0.06	1.08	3.20	8.82	30.61	0.34	0.31	4.10	8.27	27.12
80	0.24	1.43	4.66	11.00	34.27	0.00	0.88	5.40	13.76	34.87
100	0.27	1.47	6,02	14.71	41,60	0.20	1.96	6,03	17,79	40,55
120	0.00	1.10	6,27	14,92	42,05	0.13	1.74	9,24	19,68	44,45
140	0.22	2.19	9,29	18,34	48,96	0.27	2.66	8,34	21,59	48,90
160	0.13	3.32	10,32	25,49	55,35	0.19	3.03	10,26	25,39	55,32
α = 40% *and* β = 60%										
20	0.00	0.66	0.00	2.63	6.99	0.00	0.00	1.43	0.02	8.90
40	0.00	0.46	1.50	3.16	14.92	0.00	0.48	0.45	5.52	15.66
60	0.06	0.84	1.62	6.66	24.02	0.34	0.02	2.41	7.13	22.69
80	0.24	1.37	3.10	7.30	26.65	0.00	0.65	4.00	9.90	27.80
100	0.19	0.84	4,26	10,57	33,37	0.20	1.07	3,96	12,07	32,44
120	0.00	0.84	4,53	10,80	33,95	0.13	0.79	6,39	14,07	37,09
140	0.18	0.89	6,36	14,58	39,30	0.25	1.61	5,34	15,18	41,36
160	0.13	1.88	7,27	20,78	46,81	0.19	2.03	8,06	18,93	49,38

more importance to the morphology factor, improved the results (comparing the results for the same number of objects and same velocities). Results might still be improved by using different configurations of the α β parameters, so this is will be one of our future efforts in the improvement of tracking algorithms.

4.3 Cluster Tracking

The Create Clusters property was used to test the same tracking algorithms (Simple NN and NN with Morphology Algorithms with α = 40%). The simulated parameters were: number of clusters (1, 5 and 10), number of objects per cluster (10 and 15), maximum velocity (5 and 10), Alternative Movement, Center Force (4) and morphology factor (0 and 0.05). The tracking results are presented in Table 3. For the Cluster creation, we used 10 time-series (and averaged the results) of 200 frames and calculated the object tracking error on every frame accumulated throughout the time-series.

The DBSCAN algorithm tries to separate each cluster in every frame. Therefore, if the number of clusters is the same between the actual frame and the previous one (t and t − 1), then they are matched using NN, treating them as isolated objects and aligned using their centroids. If the number of clusters changes, the first step is skipped and the number of objects inside each cluster is checked. When, inside a cluster, there are more objects in t then in t − 1, these 'extra' objects are labeled as 'Possible Entry'. If there are fewer objects, they are labeled 'Possible Exit'. This tagging is temporary and

compares the "Possible Exit" features to the features of all other objects of the frame t − 1, linking it to a "Possible entry" in another cluster (meaning that it left one cluster to join another), classifying it as noise, or as an object leaving the image. The main difference between DBSCAN 1 and DBSCAN 2 algorithms is that, in the first, this classification is done after the tracking and in the second it is done before the tracking, equalizing the number of objects between the clusters.

Table 3. Tracking errors, within clusters with different properties, using the Simple and Morphology NN Algorithms with different number of clusters (1 to 10), different number of objects per cluster (5 to 15), and different maximum velocities (2 to 10).

Nº of clusters	Obj./Clusters	m factor = 0			m factor = 0.05		
		V = 2	V = 5	V = 10	V = 2	V = 5	V = 10
Simple NN Algorithm							
1	5	2.95	0.58	2.62	1.93	2.32	14.23
	10	3.32	7.79	30.42	0.93	9.88	23.33
	15	4.63	11.74	50.91	2.94	10.74	38.06
3	5	0.05	2.40	7.69	0.00	4.57	9.01
	10	1.07	7.11	27.83	2.20	9.07	30.08
	15	3.05	14.74	43.77	2.76	16.77	45.44
5	5	0.23	2.22	6.25	0.72	3.28	9.74
	10	0.70	7.48	34.71	1.57	10.95	31.89
	15	3.06	17.43	45.22	3.53	16.06	44.51
7	5	0.58	2.55	12.78	1.04	1.95	14.52
	10	1.58	11.21	33.76	1.81	11.78	40.35
	15	3.14	19.81	48.55	4.01	17.75	48.96
10	5	0.99	3.39	17.81	0.25	5.40	17.13
	10	1.95	12.20	38.26	1.52	11.64	42.47
	15	3.84	21.14	53.90	4.87	23.52	57.34
NN with morphology (α = 40% *and* β = 60%)							
1	5	0.00	0.00	0.00	1.93	0.00	7.92
	10	0.01	1.27	4.88	3.04	5.52	13.83
	15	0.18	3.76	21.14	1.75	4.63	20.76
3	5	0.38	1.08	1.66	0.00	1.23	4.08
	10	1.18	1.26	10.33	0.03	2.13	12.52
	15	1.81	5.29	20.24	1.92	8.12	22.44
5	5	0.00	1.78	2.58	0.10	0.68	5.77
	10	0.71	1.80	12.98	0.15	4.69	15.93
	15	1.54	7.16	20.77	0.93	5.95	22.07
7	5	0.20	0.41	2.82	0.41	0.35	5.13
	10	0.78	3.92	15.08	0.48	3.60	17.84
	15	1.22	8.14	25.78	1.99	6.86	27.11
10	5	0.04	0.97	6.93	0.15	2.31	7.20
	10	0.48	3.78	16.15	0.54	4.55	19.71
	15	1.11	8.73	28.36	2.22	10.13	34.12

Results from both DBSCAN Algorithms are presented in Table 4 (m factor = 0.00) and Table 5 (m factor = 0.05).

Table 4. DBSCAN1 (DB1) and DBSCAN1 (DB2) tracking errors comparison for different number of clusters, objects per cluster, and maximum velocities, with m factor = 0.

mmd = 0.00							
Clusters	Objects/Cluster	Vmax = 2		Vmax = 5		Vmax = 10	
		DB1	DB2	DB1	DB2	DB1	DB2
1	5	0.00	0.00	0.00	0.00	0.16	0.16
	10	0.00	0.00	5.55	4.67	2.97	2.97
	15	0.24	0.24	1.92	2.59	12.94	12.96
3	5	3.10	2.47	5.76	6.01	8.81	8.72
	10	0.70	1.02	4.86	4.89	11.28	10.70
	15	0.83	0.83	3.86	3.86	16.44	16.34
5	5	5.01	5.20	2.46	2.58	15.87	16.80
	10	1.44	1.04	5.43	5.54	13.71	14.56
	15	0.27	0.27	5.84	5.74	19.29	19.35
7	5	2.05	1.89	4.82	5.29	6.37	6.49
	10	2.47	2.23	4.52	4.81	16.18	16.51
	15	0.83	0.83	8.44	8.60	25.45	25.36
10	5	3.15	2.82	9.50	9.03	11.90	12.11
	10	2.45	3.33	5.81	6.00	17.55	17.57
	15	1.24	1.24	8.65	8.65	28.29	28.29

Comparing Table 3 with Table 1, we can observe that the simple NN cannot handle clusters adequately (for V = 10, m factor = 0.05 and 160 objects/cluster, we have a 4,12% error while for V = 10, m factor = 0.05, 10 clusters and 15 objects/cluster, for a total of 150 objects we have a 57.34% error rate). From Table 3, we can observe that the NNm algorithm handles much better the cluster creation, giving almost one half of the errors (worst case scenario of 34.12% versus 57.34% for the same configuration).

From Tables 4 and 5 we can conclude that DBSCAN Algorithms do not improve significantly over the NNm algorithm (Table 3) for the same configuration. A strange behavior for lower velocities was identified in both DBSCAN algorithms, where increasing the objects actually decreased the tracking errors. This behavior is explainable by the higher movement restriction of objects belonging to clusters with larger number objects, but further studies are required to further analyze this behavior. This behavior has not been identified in both simple NN and NNm algorithms.

Table 5. DBSCAN1 (DB1) and DBSCAN1 (DB2) tracking errors comparison for different number of clusters, objects per cluster, and maximum velocities, with m factor = 0.05.

mmd = 0.05							
Clusters	Objects/Cluster	Vmax = 2		Vmax = 5		Vmax = 10	
		DB1	DB2	DB1	DB2	DB1	DB2
1	5	0.96	0.96	1.67	1.67	9.75	9.75
	10	0.00	0.00	9.64	9.64	9.14	7.82
	15	0.85	0.85	3.87	3.87	10.49	10.49
3	5	0.00	0.00	5.06	5.02	13.97	14.89
	10	5.45	5.43	3.06	2.81	9.91	9.83
	15	1.99	1.99	3.78	3.77	19.55	19.63
5	5	2.18	2.69	10.72	11.23	20.79	22.28
	10	1.94	2.33	6.42	7.55	16.61	17.37
	15	0.79	0.79	6.49	6.19	20.84	20.82
7	5	4.07	4.20	7.54	7.43	13.78	12.98
	10	3.37	4.06	4.63	5.46	18.29	18.54
	15	2.00	2.00	7.06	7.22	27.59	27.70
10	5	2.02	1.90	8.33	8.41	13.55	14.38
	10	1.81	2.30	6.42	6.43	21.07	21.42
	15	2.76	2.67	9.91	9.98	34.52	34.52

5 Conclusions and Future Work

To support high-throughput experiments of single cell imaging, reliable automated image processing methods are required. Although most studies focus on automatic segmentation of cells or cellular structures, in time-series proper object tracking is also necessary, especially because tracking errors propagate, meaning that even small tracking errors (particularly on the initial frames) lead to a high percentage of misidentified tracks overall.

To validate Tracking Algorithms, it is necessary to use a labelled 'ground truth'. Sometimes this ground-truth can be manually obtained, but this strategy is not feasible on a Big Data scenario. A more viable alternative is to generate artificial images by simulating biological cell models. To produce such artificial images, we developed an open source platform that can simulate biologically inspired bacterial systems, by creating cells that of different shapes and sizes, cells that can grow and divide and, cells that can move as a single objects or as clustered objects.

Using this Platform, we evaluated three tracking algorithms (Simple NN, NNm and two variations of the DBSCAN Algorithm). The obtained results show that, for cases with lower maximum velocity, the Simple NN Algorithm was able to track objects even with a significant increase in the number of objects.

Meanwhile, the NNm algorithm can help reducing tracking errors when the velocity is increased. In the example where we forced the creation of clusters, the Simple NN algorithm was unable to handle the increase of number of clusters and objects in a cluster (even for a constant number of objects). On the other hand, the

NNm and the DBSCAN algorithms showed similar, significant capabilities to handle large clusters. In the near future, we plan to study and compare other tracking methodologies in different cluster configurations using the proposed framework. Here, the newly developed object division module will be used to test division tracking in dense clusters.

We expect this open-sourced tool[1] to help future endeavors in the development of new tracking algorithms, as it can produce huge amounts of benchmarked images.

The next steps of our work will be to introduce a new module that generates secondary bodies inside the primary objects, simulating internal cell organelles and structures. A future application will also be made available as a web-based system to improve usability and compatibility.

Acknowledgments. Work supported by the Portuguese Foundation for Science and Technology (FCT/MCTES) through a PhD Scholarship, ref. SFRH/BD/88987/2012 to LM, SADAC project (ref. PTDC/BBB-MET/1084/2012) and by FCT Strategic Program UID/EEA/00066/203 of UNINOVA, CTS. This work is also funded by the Academy of Finland [refs. 295027 and 305342 to ASR] and the Jane and Aatos Erkko Foundation [ref. 5-3416-12 to ASR].

References

1. Danuser, G.: Computer vision in cell biology. Cell **147**, 973–978 (2011)
2. Sung, M.-H., McNally, J.G.: Live cell imaging and systems biology. Wiley Interdiscip. Rev. Syst. Biol. Med. **3**, 167–182 (2011)
3. Coutu, D.L., Schroeder, T.: Probing cellular processes by long-term live imaging–historic problems and current solutions. J. Cell Sci. **126**, 3805–3815 (2013)
4. Bonnet, N.: Some trends in microscope image processing. Micron **35**, 635–653 (2004)
5. Frigault, M., Lacoste, J., Swift, J., Brown, C.: Live-cell microscopy - tips and tools. J. Cell Sci. **122**, 753–767 (2009)
6. Deshmukh, M., Bhosle, U.: A survey of image registration. Int. J. Image Process. **5**, 245–269 (2011)
7. Wyawahare, M., Patil, P., Abhyankar, H.: Image registration techniques: an overview. Int. J. Signal Process. Image Process Pattern Recognit. **2**, 11–28 (2009)
8. Meijering, E.: Cell segmentation: 50 years down the road. IEEE Sig. Process. Mag. **29**, 140–145 (2012)
9. Tissainayagam, P., Suter, D.: Object tracking in image sequences using point features. Pattern Recognit. **38**, 105–113 (2005)
10. Yilmaz, A., Javed, O., Shah, M.: Object tracking: a survey. ACM Comput. Surv. **38**, 1–45 (2006)
11. Selinummi, J., Seppälä, J., Yli-Harja, O., Puhakka, J.: Software for quantification of labeled bacteria from digital microscope images by automated image analysis. Biotechniques **39**, 859–863 (2005)
12. Wang, Q., Niemi, J., Tan, C.-M., You, L., West, M.: Image segmentation and dynamic lineage analysis in single-cell fluorescence microscopy. Cytom. A. **77**, 101–110 (2010)

[1] Tool available at: http://griduni.uninova.pt/Clustergen/ ClusterGen_v1.0.zip.

13. Sliusarenko, O., Heinritz, J.: High-throughput, subpixel precision analysis of bacterial morphogenesis and intracellular spatio-temporal dynamics. Mol. Microbiol. **80**, 612–627 (2011)
14. Young, J., Locke, J.C.W., Altinok, A., Rosenfeld, N., Bacarian, T., Swain, P.S., Mjolsness, E., Elowitz, M.B.: Measuring single-cell gene expression dynamics in bacteria using fluorescence time-lapse microscopy. Nat. Protoc. **7**, 80–88 (2012)
15. Häkkinen, A., Muthukrishnan, A.-B., Mora, A., Fonseca, J.M., Ribeiro, A.S.: Cell Aging: a tool to study segregation and partitioning in division in cell lineages of Escherichia coli. Bioinformatics **29**, 1708–1709 (2013)
16. Coelho, L.P., Shariff, A., Murphy, R.F.: Nuclear segmentation in microscope cell images a hand-segmented dataset and comparison of algorithms. In: Proceedings of IEEE International Symposium on Biomedical Imaging, pp. 518–521 (2009)
17. Xiong, W., Wang, Y., Ong, S.H., Lim, J.H., Jiang, L.: Learning cell geometry models for cell image simulation : an unbiased approach. In: Proceedings of 2010 IEEE 17th International Conference on Image Processing, pp. 1897–1900 (2010)
18. Kruse, K.: Bacterial organization in space and time. In: Comprehensive Biophysics, pp. 208–221 (2012)
19. Misteli, T.: Beyond the sequence: cellular organization of genome function. Cell **128**, 787–800 (2007)
20. Svoboda, D., Kozubek, M., Stejskal, S.: Generation of digital phantoms of cell nuclei and simulation of image formation in 3D image cytometry. Cytometry. A. **75**, 494–509 (2009)
21. Lehmussola, A., Ruusuvuori, P., Selinummi, J., Huttunen, H., Yli-Harja, O.: Computational framework for simulating fluorescence microscope images with cell populations. IEEE Trans. Med. Imag. **26**, 1010–1016 (2007)
22. Ruusuvuori, P., Lehmussola, A., Selinummi, J., Rajala, T., Huttunen, H., Yli-Harja, O.: Benchmark set of synthetic images for validating cell image analysis algorithms. In: Proceedings of the 16th European Signal Processing Conference, EUSIPCO (2008)
23. Lehmussola, A., Ruusuvuori, P., Selinummi, J., Rajala, T., Yli-harja, O.: Synthetic images of high-throughput microscopy for validation of image analysis methods. Proc. IEEE **96**, 1348–1360 (2011)
24. Svoboda, D., Kasik, M., Maska, M., Hubeny, J.: On simulating 3D fluorescent microscope images. In: Proceedings of 12th International Conference on Computer Analysis of Images and Patterns, CAIP 2007, Vienna, Austria, 27–29 August 2007, pp. 309–316 (2007)
25. Ulman, V., Oremus, Z., Svoboda, D.: TRAgen: a tool for generation of synthetic time-lapse image sequences of living cells. In: Proceedings of 18th International Conference on Image Analysis and Processing (ICIAP 2015), pp. 623–634. Springer (2015)
26. Satwik, R., Benjamin, P., Nicholas, H., Steven, A., Lani, W.: SimuCell: a flexible framework for creating synthetic microscopy images a PhenoRipper: software for rapidly profiling microscopy images. Nat. Meth. **9**, 634–636 (2012)
27. Murphy, R.: Cell Organizer: image-derived models of subcellular organization and protein distribution. Meth. Cell Biol. **110**, 179–193 (2012)
28. Zhao, T., Murphy, R.F.: Automated learning of generative models for subcellular location: building blocks for systems biology. Cytometry. A. **71**, 978–990 (2007)
29. Martins, L., Fonseca, J., Ribeiro, A.: "miSimBa" - a simulator of synthetic time-lapsed microscopy images of bacterial cells. In: Proceedings of 2015 IEEE 4th Portuguese Meeting on Bioengineering, ENBENG 2015, pp. 1–6 (2015)
30. Gotelli, N.J., McGill, B.J.: Null versus neutral models: what's the difference? Ecography (Cop.) **29**, 793–800 (1996)
31. Elfring, J., Janssen, R., van de Molengraft, R.: Data association and tracking: a literature survey. In: ICT Call 4 RoboEarth Project (2010)

32. Gu, S., Zheng, Y., Tomasi, C.: Efficient visual object tracking with online nearest neighbor classifier. In: Computer Vision – ACCV 2010. LNCS, vol. 6492, pp. 271–282 (2011)
33. Gorji, A., Menhaj, M.B.: Multiple target tracking for mobile robots using the JPDAF algorithm. In: 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007), pp. 137–145 (2007)
34. Gordon, N., Salmond, D., Smith, A.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. Radar Sig. Process. IEE Proc. F. **140**, 107–113 (1993)
35. Bhattacharyya, A.: On a measure of divergence between two statistical populations defined by probability distributions. Bull. Calcutta Math. Soc. **35**, 99–110 (1943)
36. Joyce, J.: Kullback-Leibler Divergence. In: Lovric, M. (ed.) International Encyclopedia of Statistical Science SE - 327, pp. 720–722. Springer, Heidelberg (2014)
37. Zhou, H., Yuan, Y., Shi, C.: Object tracking using SIFT features and mean shift. Comput. Vis. Image Underst. **113**, 345–352 (2009)
38. Shi, J., Tomasi, C.: Good features to track. In: 1994 IEEE Computer Society Conference on CVPR 1994, pp. 593–600. IEEE (1994)
39. Cabeen, M.T., Jacobs-Wagner, C.: Bacterial cell shape. Nat. Rev. Microbiol. **3**, 601–610 (2005)
40. Salton, M., Kim, K.: Structure. In: Baron, S. (ed.) Medical Microbiology, 4th edn., Chap. 2. University of Texas Medical Branch at Galveston, Galveston (1996)
41. Zinder, S.H., Dworkin, M.: Morphological and physiological diversity. In: Dworkin, M., Falkow, S., Rosenberg, E., Schleifer, K.-H., Stackebrandt, E. (eds.) Prokaryotes, Chap. 1.7, pp. 185–220. Springer, New York (2006)
42. Koch, A.L.: What size should a bacterium be? A question of scale. Annu. Rev. Microbiol. **50**, 317–348 (1996)
43. Höltje, J.-V.: Cell walls, bacterial. In: The Desk Encyclopedia of Microbiology, pp. 239–250 (2004)
44. Henning, U., Rehn, K., Hoehn, B.: Cell envelope and shape of Escherichia coli K12. Proc. Natl. Acad. Sci. USA **70**, 2033–2036 (1973)
45. Carballido-López, R., Formstone, A.: Shape determination in Bacillus subtilis. Curr. Opin. Microbiol. **10**, 611–616 (2007)
46. Höltje, J.-V.: Growth of the stress-bearing and shape-maintaining murein sacculus of Escherichia coli. Microbiol. Mol. Biol. Rev. **62**, 181–203 (1998)
47. Huang, K.C., Mukhopadhyay, R., Wen, B., Gitai, Z., Wingreen, N.S.: Cell shape and cell-wall organization in Gram-negative bacteria. Proc. Natl. Acad. Sci. U.S.A. **105**, 19282–19287 (2008)
48. Canelas, P., Martins, L., Mora, A., Ribeiro, A.S., Fonseca, J.: An image generator platform to improve cell tracking algorithms - simulation of objects of various morphologies, kinetics and clustering. In: Proceedings of the 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, pp. 44–55 (2016). ISBN 978-989-758-199-1
49. Wang, J.D., Levin, P.A.: Metabolism, cell growth and the bacterial cell cycle. Nat. Rev. Microbiol. **7**, 822–827 (2009)
50. Young, K.D.: Bacterial shape: two-dimensional questions and possibilities. Annu. Rev. Microbiol. **64**, 223–240 (2010)
51. Zapun, A., Vernet, T., Pinho, M.: The different shapes of cocci. FEMS Microbiol. Rev. **32**, 345–360 (2008)
52. Lauffenburger, D.: Effects of cell motility and chemotaxis on microbial population growth. Biophys. J. **40**, 209–219 (1982)

53. Czink, N., Mecklenbräuker, C., Del Galdo, G.: A novel automatic cluster tracking algorithm. In: 2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, pp. 1–5 (2006)
54. Tran, T.N., Drab, K., Daszykowski, M.: Revised DBSCAN algorithm to cluster data with dense adjacent clusters. Chemom. Intell. Lab. Syst. **120**, 92–96 (2013)
55. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231 (1996)

Proteins Flexibility as a Criterion for Elucidation of Activating Mutants in Personalized Cancer Medicine

Igor F. Tsigelny[1,2,3,5(✉)], Razelle Kurzrock[1], Åge Aleksander Skjevik[2,4], Valentina L. Kouznetsova[1,2], Amélie Boichard[1], and Sadakatsu Ikeda[1]

[1] Moores Cancer Center, University of California, San Diego, La Jolla, San Diego, CA, USA
{itsigeln, rkurzrock, vkouznetsova, saikeda}@ucsd.edu
[2] San Diego Supercomputer Center, University of California, San Diego, La Jolla, San Diego, CA, USA
age.a.skjevik@gmail.com
[3] Department of Neurosciences, University of California, San Diego, La Jolla, San Diego, CA, USA
[4] Department of Biomedicine, University of Bergen, Bergen, Norway
[5] CureMatch Inc., San Diego, CA, USA

Abstract. We developed a new strategy for elucidation of functional impact of mutations in proteins. Using molecular dynamics simulations, we explore flexibility of proteins in the sites of their binding to the other proteins. Binding of two or more proteins go through the stage of intermediate binding complexes. On this stage the number of possible conformations of the proteins' binding sites are interacting with each other. Increasing flexibility in the binding sites increase a probability of the best-energy docking of proteins. Our computational simulations demonstrated that a missense alteration of MET (p.Tyr501Cys), which lead to an increase of flexibility of the protein, may improve the binding of the receptor with its ligand HGF (hepatocyte growth factor) and thus be considered as activating. Accordingly to this conclusion, a patient presenting a hepatocellular carcinoma MET Y501C-mutated showed a good response when treated by a potent MET inhibitor (cabozantinib), with a decrease of −65% of the alpha-foeto-protein (AFP).

Keywords: Personalized cancer medicine · Molecular dynamics · Structure of proteins · Sema domain · MET · c-Met

1 Introduction

In Personalized Medicine, physicians select therapeutic regimens based on patients and diseases unique features, such as genomic or proteomic markers. In oncology particularly, treatments need to counteract the molecular alterations driving the tumor progression [16]. Nevertheless, one characteristics of tumor cells is their lack of genomic stability, which usually results in the accumulation of a high number of mutations [3].

M.S. Obaidat et al. (eds.), *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing 676,
https://doi.org/10.1007/978-3-319-69832-8_5

Long before the wide availability of high-throughput molecular technics, the average number of mutations presented by a tumor was evaluated to nearly 10,000 alterations by cell [33, 35]; and the development of next-generation DNA-sequencing methods only reinforced the hypothesis of a high genomic complexity [39]. Launched in 2004, the Catalogue of Somatic Mutations in Cancer (COSMIC) is a database aimed to collect and display expert-curated information on acquired mutations found in human cancers [1]; (http://cancer.sanger.ac.uk/cosmic). In 2016, this repository included more than 4 million coding mutations across the entire genome, most of them located within 400 key cancer genes [8].

One of the most important task for Personalized Medicine is to elucidate, at the protein level, the functional effect of each of the genomic variations [25, 31], decipher the molecular determinants of drug-specific sensitivity in tumors [37] and, finally, predict the outcome of patients treated by such molecular-driven therapies. In certain cases, the structural and/or functional impact of a single residue change within a protein sequence may be simple to estimate, for example, when the change occurs between two residues oppositely charged and implicated in a salt bridge, or by insertion of a hydrophobic residue instead of a hydrophilic residue that participates in hydrogen bonding [34]. However, in other cases, the effect of a residue substitution might not be that obvious. In this chapter, we will review a possibility to determine protein activity changes that occur due to these mutations and show how molecular dynamics (MD) simulation might help in this concern.

2 MET Structure and Function

MET (also called hepatocyte growth factor receptor, HGFR) is a tyrosine kinase receptor encoded by the proto-oncogene MET.

MET primary product is a pre-pro-protein, which undergoes a proteolysis to generate alpha and beta subunits, further linked via disulfide bonds to form the mature receptor. The structure of the mature receptor includes an extracellular region, composed of a semaphorins-homology domain (Sema domain), a cysteine-rich domain (MRS domain), and four immunoglobulin-like structures (Ig domains); and a juxtamembrane/intracellular region responsible for the MET tyrosine kinase activity (kinase domain) [9]. The MET receptor has significant structural similarity to Ron and Sea receptors [12, 27, 32].

Binding of MET with its ligand HGF (hepatocyte growth factor), occuring within the Sema domain, induces the dimerization and activation of the receptor and promotes cellular survival, migration, and invasion [36]. This cell-surface receptor is expressed in epithelial cells of many organs including the liver, pancreas, prostate, kidney, muscle, and bone marrow, during both embryogenesis and adulthood [21]. MET-activating point mutations have been associated with several tumor types: hepatocellular carcinoma, head and neck cancers, and papillary renal cell carcinoma [40]. Additional alterations, such as amplification or overexpression, have also been related to the occurrence of multiple human cancers [6, 14, 15, 20, 24].

In order to be activated, MET requires binding of its ligand HGF, which is active only after proteolytic conversion to a two-chain configuration [11, 32]. Direct binding

sites show that HGF-beta chains bind to the extracellular domain of MET with a Kd of about 90 nM. Analysis of the MET–HGF interface shows a set of moderately complementary side chains on both sides (Fig. 1).

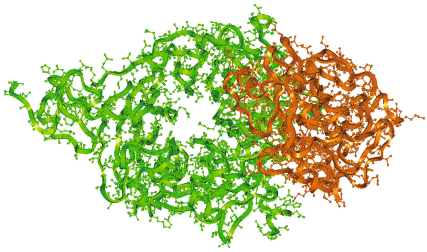


Fig. 1. Interaction of the extracellular Sema domain of Met (green) with the HGF-beta chain (brown).

2.1 Sema Domain and MET p.Tyr501Cys Point Mutation

The Sema domain of MET forms a "seven-bladed beta-propeller" having a shape of a funnel with an inner diameter of 25 Å and a total diameter of approximately 50 Å. The blades of this propeller are antiparallel beta-strands. The Sema domain is stabilized by the interactions between C- and N-terminal residues, and the beta-propeller structure is stabilized by seven disulfide bridges also found in a number of proteins with notable amino-acid homology. The HGF beta-chain associates with the Sema domain at the bottom face of the propeller with at least seven electrostatic pair interactions between the two proteins [32, 38].

In this chapter, we report an unusual mutation (observed in a patient carrying a hepatocellular carcinoma) located within the residue tyrosine 501 of MET (p.Tyr501Cys), which corresponds to the interface between the C- and N-terminal of the Sema domain. From the general point of view and using basics of amino-acid physicochemistry features and polypeptide structure and conformation, substitution of a tyrosine residue to a cysteine residue would unlikely make any changes, unless said cysteine creates a new disulfide bond, which is not the case here. In the chapter, we hypothesized that this particular mutation involving a codon, which participates in the hydrophobic node linking the N- and C-terminals of the Sema domain, may affect the flexibility and thus the capacity of activation of the MET receptor after binding to its ligand.

3 Flexibility Defines Protein-Protein Binding

Creation of the MET–HGF complex is defined by docking of these two proteins. It has been shown by Panjkovich and Daura [22] that flexibility in the specific binding points of proteins is a critical parameter that affects their binding. The authors elucidated proteins' flexibility using Normal Mode Analysis (NMA), which calculated low-frequency modes that correspond to large collective oscillations of the protein and high-frequency modes that correspond to small local fluctuations (as described by Dykeman and Twarock [7], and what are most interesting, a flexibility value can be a

criterion of protein-docking possible “hot spots”. Marsh and Teichmann (2014) studied the effect of protein flexibility on evolution of protein–protein complexes. They found the more nonhomologous subunits assemble into the complex, the more flexible they have to be. Liu and coauthors demonstrated that crystallographic B-factors, which are related to vibrational motion of protein atoms, are an important feature that can elucidate the biological interface in protein–protein complexes [18]. They also showed that B-factors could separate the real biologic interfaces in complexes from artificial interfaces created by the crystal packing (i.e., in preparation of crystals for X-ray crystallography). B-factors are used for prediction of protein–protein interfaces in the program SPIDER [23]. Levy and colleagues studied more than 100 protein–protein complexes and demonstrated that flexibility of the binding partners is a very important feature of protein–protein association [17]. Actually, the binding process is not only related to the three-dimensional structure of proteins but also to the four-dimensional set of possible conformations adopted by means of the flexible regions of the involved proteins. Transition-state conformational ensembles for general folding of proteins and their bindings have similar characteristics for different proteins [17]. Taking into consideration importance of local flexibility of protein–protein binding sites, we assumed that increasing flexibility of the Sema domain of MET would improve its binding with HGF and consequently increase the activity of the entire complex.

4 Mutation Y501C Increases Flexibility of the Sema Domain in Its Binding Sites

We conducted MD for the wild-type and mutated proteins. We found significant increase in flexibility of several binding sites of the Sema domain (Fig. 2). The violet rectangles encompass the most flexible regions of the mutant structure. Maximum flexibility changes occur in the binding sites of the Sema domain that are in direct contact with the HGF protein (Fig. 3). Note that the regions around residues 150 and 209 of the Sema domain having profound picks on the plot are in direct contact with

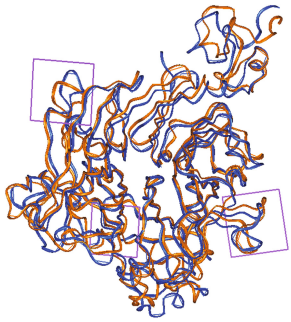


Fig. 2. Superposition of the 300 ns conformers, with the wild-type represented by brown ribbons of the alpha-trace and the Y501C mutant of the Sema domain of MET shown as blue ribbons [38].

HGF residues during its binding. These results suggest that the mutation Y501C (tyrosine to cysteine) of MET leads to a significant increase in the flexibility of the MET Sema domain in the regions contacting HFG. Such changes may improve the binding and consequently the activity of the MET–HGF complex.

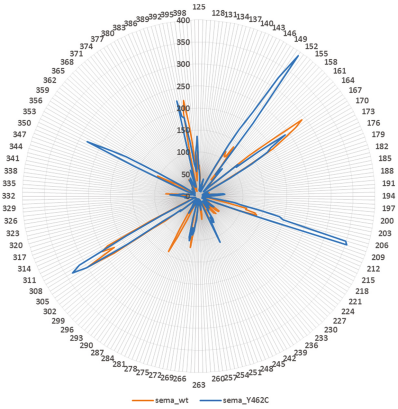


Fig. 3. Flexibility (defined by B-factor values) of the Sema domain residues in the wild-type (brown) and Y501C mutant (blue) conformers as calculated from a 300 ns MD trajectory. Residues numbers are on the circle.

5 Patient Treatment Based on the MD Simulations

A patient was diagnosed with hepatocellular carcinoma (HCC), with the MET Y501C missense mutation that was elucidated from circulating tumor DNA. Using results of MD simulation, we estimated that this mutation activates MET protein that is active in cancer development. The drug cabozantinib—a MET inhibitor was prescribed for the patient. This drug caused significant (65%) reduction of alpha-pheto protein (AFP), a tumor marker for HCC.

6 Conclusion

Study of flexibility of binding sites in protein–protein complexes with molecular dynamics simulations can be used for estimation of possible functional impact of mutations of amino acids located in these sites. This information is extremely important to physicians for making decision on proper treatment.

7 Methods

Sema domain mutant Y462C was produced by replacement tyrosine 462 with cysteine. Disulfide bridges between the relevant cysteine pairs in each protein were generated. The SEMA domain of MET protein from the complex with heparin [32] pdb ID 1shy was used.

All simulations were conducted as described by Tsigelny and coauthors, starting with molecular dynamics (MD) for both the wild-type and mutated versions of Sema domain of MET [38]. Each of proteins were placed in an octahedral water box consisting of approximately 48,500 TIP3P water molecules and 9 neutralizing K^+ ions modeled by Joung/Cheatham ion parameters [13]. The simulations were conducted using the GPU/CUDA-accelerated version of PMEMD implemented in the AMBER14 software suite [4, 5, 10, 29, 30], with the protein described by AMBER ff14 SB parameters. Each of the two protein systems were subjected to the following minimization/simulation steps: (i) Unrestrained minimization for 10,000 steps; (ii) Gradual constant volume heating from 0 to 100 K over 5 ps with restraints applied to the protein backbone; (iii) Gradual constant pressure heating to 310 K over 100 ps with restraints applied to the protein backbone; (iv) 300 ns unrestrained constant pressure simulation at 310 K.

The Langevin thermostat algorithm [19] was applied for regulation of temperature with a 1.0 ps^{-1} collision frequency, and the pressure was regulated isotropically during the second heating step and the production simulation by means of the Berendsen barostat algorithm [2] at a reference pressure of 1.0 bar. Bond lengths for bonds involving hydrogen were constrained using the SHAKE algorithm [28], allowing for a time step of 2 fs. Periodic boundary conditions were applied, and the particle mesh Ewald (PME) method [26] was used for the evaluation of electrostatics.

References

1. Bamford, S., Dawson, E., Forbes, S., Clements, J., Pettett, R., Dogan, A., Flanagan, A., Teague, J., Futreal, P.A., Stratton, M.R., Wooster, R.: The COSMIC (Catalogue of Somatic Mutations in Cancer) database and website. Br. J. Cancer **91**(2), 355–358 (2004)
2. Berendsen, H.J.C., Postma, J.P.M., van Gunsteren, W.F., DiNola, A., Haak, J.R.: Molecular dynamics with coupling to an external bath. J. Chem. Phys. **81**, 3684–3690 (1984)
3. Bozic, I., Antal, T., Ohtsuki, H., Carter, H., Kim, D., Chen, S., Karchin, R., Kinzler, K.W., Vogelstein, B., Nowak, M.A.: Accumulation of driver and passenger mutations during tumor progression. Proc. Natl. Acad. Sci. U.S.A. **107**(43), 18545–18550 (2010)
4. Case, D.A., Babin, V., Berriman, J.T., et al.: AMBER 14. University of California, San Francisco (2014)
5. Darden, T., York, D., Pedersen, L.: Particle mesh Ewald: an Nlog(N) method for Ewald sums in large systems. J. Chem. Phys. **98**, 10089–10092 (1993)
6. Di Renzo, M.F., Narsimhan, R.P., Olivero, M., Bretti, S., Giordano, S., Medico, E., Gaglia, P., Zara, P., Comoglio, P.M.: Expression of the Met/HGF receptor in normal and neoplastic human tissues. Oncogene **6**(11), 1997–2003 (1991)
7. Dykeman, E.C., Twarock, R.: All-atom normal-mode analysis reveals an RNA-induced allostery in a bacteriophage coat protein. Phys. Rev. E Stat. Nonlin. Soft Matter Phys. **81**(3 Pt. 1), 031908 (2010)

8. Forbes, S.A., Beare, D., Bindal, N., Bamford, S., Ward, S., Cole, C.G., Jia, M., Kok, C., Boutselakis, H., De, T., Sondka, Z., Ponting, L., Stefancsik, R., Harsha, B., Tate, J., Dawson, E., Thompson, S., Jubb, H., Campbell, P.J.: COSMIC: high-resolution cancer genetics using the catalogue of somatic mutations in cancer. Curr. Protoc. Hum. Genet. **91**, 10.11.1–10.11.37 (2016)
9. Gherardi, E., Youles, M.E., Miguel, R.N., Blundell, T.L., Iamele, L., Gough, J., Bandyopadhyay, A., Hartmann, G., Butler, P.J.G.: Functional map and domain structure of MET, the product of the c-Met protooncogene and receptor for hepatocyte growth factor/scatter factor. Proc. Natl. Acad. Sci. U.S.A. **100**(21), 12039–12044 (2003)
10. Götz, A.W., Williamson, M.J., Xu, D., Poole, D., Le Grand, S., Walker, R.C.: Routine microsecond molecular dynamics simulations with AMBER on GPUs. 1. Generalized born. J. Chem. Theory Comput. **8**, 1542–1555 (2012)
11. Hartmann, G., Naldini, L., Weidner, K.M., Sachs, M., Vigna, E., Comoglio, P.M., Birchmeier, W.: A functional domain in the heavy chain of scatter factor/hepatocyte growth factor binds the c-Met receptor and induces cell dissociation but not mitogenesis. Proc. Natl. Acad. Sci. U.S.A. **89**, 11574–11578 (1992)
12. Huff, J.L., Jelinek, M.A., Borgman, C.A., Lansing, T.J., Parsons, J.T.: The protooncogene c-sea encodes a transmembrane proteintyrosine kinase related to the Met/hepatocyte growth factor/scatter factor receptor. Proc. Natl. Acad. Sci. U.S.A. **90**, 6140–6144 (1993)
13. Joung, I.S., Cheatham III, T.E.: Determination of alkali and halide monovalent ion parameters for use in explicitly solvated biomolecular simulations. J. Phys. Chem. B **112**, 9020–9041 (2008)
14. Jücker, M., Günther, A., Gradl, G., Fonatsch, C., Krueger, G., Diehl, V., Tesch, H.: The Met/hepatocyte growth factor receptor (HGFR) gene is overexpressed in some cases of human leukemia and lymphoma. Leuk. Res. **18**(1), 7–16 (1994)
15. Kawakami, H., Okamoto, I., Okamoto, W., Tanizaki, J., Nakagawa, K., Nishio, K.: Targeting MET amplification as a new oncogenic driver. Cancers **6**(3), 1540–1552 (2014)
16. Kurzrock, R., Giles, F.J.: Precision oncology for patients with advanced cancer: the challenges of malignant snowflakes. Cell Cycle (Georgetown, Tex.) **14**(14), 2219–2221 (2015)
17. Levy, Y., Cho, S.S., Onuchic, J.N., Wolynes, P.G.: A survey of flexible protein binding mechanisms and their transition states using native topology based energy landscapes. J. Mol. Biol. **346**(4), 1121–1145 (2005)
18. Liu, Q., Li, Z., Li, J.: Use B-factor related features for accurate classification between protein binding interfaces and crystal packing contacts. BMC Bioinform. **15**(Suppl. 16), S3 (2014)
19. Loncharich, R.J., Brooks, B.R., Pastor, R.W.: Langevin dynamics of peptides: the frictional dependence of isomerization rates of N-acetylalanyl-N'-methylamide. Biopolymers **32**, 523–535 (1992)
20. Montesano, R., Matsumoto, K., Nakamura, T., Orci, L.: Identification of a fibroblast-derived epithelial morphogen as hepatocyte growth factor. Cell **67**, 901–908 (1991)
21. Organ, S.L., Tsao, M.-S.: An overview of the c-MET signaling pathway. Ther. Adv. Med. Oncol. **3**(Suppl. 1), S7–S19 (2011)
22. Panjkovich, A., Daura, X.: Exploiting protein flexibility to predict the location of allosteric sites. BMC Bioinform. **2012**(13), 273 (2012)
23. Porollo, A., Meller, J.: Prediction-based fingerprints of protein-protein interactions. Proteins **66**, 630–645 (2007)
24. Prat, M., Narsimhan, R.P., Crepaldi, T., Nicotra, M.R., Natali, P.G., Comoglio, P.M.: The receptor encoded by the human c-MET oncogene is expressed in hepatocytes, epithelial cells and solid tumors. Int. J. Cancer **49**(3), 323–328 (1991)

25. Reva, B., Antipin, Y., Sander, C.: Predicting the functional impact of protein mutations: application to cancer genomics. Nucleic Acids Res. **39**(17), e118 (2011)
26. Roe, D.R., Cheatham III, T.E.: PTRAJ and CPPTRAJ: software for processing and analysis of molecular dynamics trajectory data. J. Chem. Theory Comput. **9**, 3084–3095 (2013)
27. Ronsin, C., Muscatelli, F., Mattei, M.G., Breathnach, R.: A novel putative receptor protein tyrosine kinase of the met family. Oncogene **8**, 1195–1202 (1993)
28. Ryckaert, J.-P., Ciccotti, G., Berendsen, H.J.C.: Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. J. Comput. Phys. **23**, 327–341 (1977)
29. Salomon-Ferrer, R., Case, D.A., Walker, R.C.: An overview of the Amber biomolecular simulation package. WIREs Comput. Mol. Sci. **3**, 198–210 (2013)
30. Salomon-Ferrer, R., Götz, A.W., Poole, D., Le Grand, S., Walker, R.C.: Routine microsecond molecular dynamics simulations with AMBER on GPUs. 2. Explicit solvent particle mesh Ewald. J. Chem. Theory Comput. **9**, 3878–3888 (2013)
31. Santarpia, L., Bottai, G., Kelly, C.M., Győrffy, B., Székely, B., Pusztai, L.: Deciphering and targeting oncogenic mutations and pathways in breast cancer. Oncologist **21**(9), 1063–1078 (2016)
32. Stamos, J., Lazarus, R.A., Yao, X., Kirchhofer, D., Wiesmann, C.: Crystal structure of the HGF beta-chain in complex with the Sema domain of the Met receptor. EMBO **23**, 2325–2335 (2004)
33. Stoler, D.L., Chen, N., Basik, M., Kahlenberg, M.S., Rodriguez-Bigas, A., Petrelli, N.J., Anderson, G.R.: The onset and extent of genomic instability in sporadic colorectal tumor progression. Proc. Natl. Acad. Sci. U.S.A. **96**(26), 15121–15126 (1999)
34. Studer, R.A., Dessailly, B.H., Orengo, C.A.: Residue mutations and their impact on protein structure and function: detecting beneficial and pathogenic changes. Biochem. J. **449**(3), 581–594 (2013)
35. Tomlinson, I., Sasieni, P., Bodmer, W.: How many mutations in a cancer? Am. J. Pathol. **160**(3), 755–758 (2002)
36. Trusolino, L., Comoglio, P.M.: Scatter-factor and semaphorin receptors: cell signalling for invasive growth. Nat. Rev. Cancer **2**(4), 289–300 (2002)
37. Tsigelny, I.F., Wheler, J.J., Greenberg, J.P., Kouznetsova, V.L., Stewart, D.J., Bazhenova, L., Kurzrock, R.: Molecular determinants of drug-specific sensitivity for Epidermal Growth Factor Receptor (EGFR) exon 19 and 20 mutants in non-small cell lung cancer. Oncotarget **6**, 6029–6039 (2015)
38. Tsigelny, I.F., Kurzrock, R., Skjevik, Å.A., Kouznetsova, V.L., Ikeda, S.: Molecular dynamics use in personalized cancer medicine: example of MET Y501C mutation. In: Proceedings of the 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, Lisbon, Portugal, 29–31 July 2016, pp. 71–74 (2016)
39. Vogelstein, B., Papadopoulos, N., Velculescu, V.E., Zhou, S., Diaz, L.A., Kinzler, K.W.: Cancer genome landscapes. Science **339**(6127), 1546–1558 (2013)
40. Zenali, M., deKay, J., Liu, Z., Hamilton, S., Zuo, Z., Lu, X., Bakkar, R., Mills, G., Broaddus, R.: Retrospective review of MET gene mutations. Oncoscience **2**(5), 533–541 (2015)

Distributed PowerShell Load Generator (D-PLG): A Tool for Generating Dynamic Network Traffic

Paul Jordan[1(✉)], Donald Van Patten[1], Gilbert Peterson[1], and Andrew Sellers[2]

[1] Air Force Institute of Technology, 2950 Hobson Way, Dayton, OH, USA
{paul.joran,donald.vanpatten,gilbert.peterson}@afit.edu
[2] United States Air Force Academy, 2354 Fairchild Drive,
Colorado Springs, CO, USA
andrew.sellers@usafa.edu
http://www.afit.edu
http://www.usafa.edu

Abstract. Obtaining data for the training of failure prediction algorithms has long been an issue. A framework for automating the generation of this data for the training and deployment of these algorithms has recently been introduced. Unfortunately, the framework was only tested on a single deprecated operating system. In order to generalize the approach a few key functions must be performed, one of which being realistic workload generation. Unfortunately, a workload generator capable of generating sufficient workload has not been developed for a Microsoft Windows active directory environment. This paper introduces a tool that makes the implementation of this new framework possible on a modern Microsoft operating system. We present data generated by the tool to demonstrate its efficacy, and finish with several extensions and applications.

Keywords: Network traffic generator · Load generator · Machine learning · Online failure prediction.

1 Introduction

Many ways of generating realistic traffic or capturing live traffic for replay exist. Unfortunately, most of these methods involve naively replaying previously recorded traffic which cannot successfully simulate encrypted authentication sessions. Being able to simulate this type of encrypted traffic is necessary to create realistic workload for modern identity management services. This research is the result of an ongoing attempt to generalize the Adaptive Failure Prediction

The views expressed herein are solely those of the authors and do not reflect the official policy or position of the U.S. Air Force, the Department of Defense, or the U.S. Government.

M.S. Obaidat et al. (eds.), *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing 676,
https://doi.org/10.1007/978-3-319-69832-8_6

(AFP) framework developed by Irrera et al. in [1] to predict failure in a Microsoft domain controller.

AFP automates the process of retraining a failure prediction algorithm after an underlying system change by placing a target system under load before injecting software faults to accelerate failure. Consequently, in order to produce realistic workload for an authentication service, full-stack encrypted sessions are necessary.

This work introduces the Distributed PowerShell Load Generator (D-PLG), developed for generating several kinds for network traffic in modern Microsoft Windows systems. This work also demonstrates the validity and generalizability of D-PLG by presenting results of several tests in a simulated production environment. Finally, this work demonstrates how the AFP framework leverages D-PLG in order to generate network transactions of arbitrary arity between unbounded network components with dynamic volume, variety, veracity, and velocity.

2 Related Work

The following two subsections briefly summarize recent advances in the two fields relevant to this research: online failure prediction, and network traffic generation. Specifically, how D-PLG fits into these fields.

2.1 Online Failure Prediction

Salfner et al. [2] published a survey of online failure prediction techniques that categorized machine learning approaches to failure prediction into a taxonomy. Unfortunately, these techniques require steady system states to be effective which has become increasingly difficult due to the shrinking software development lifecycle.

It has recently been pointed out that while many effective techniques for predicting failure exist, these techniques are too difficult to maintain and consequently are not being used. In response, Irrera et al. [1] introduced a framework called the Adaptive Failure Prediction (AFP) framework for dealing with this problem. The AFP automates the process of generating failure data in order to train a failure prediction algorithm. By automating this process, it can be done regularly to enable a predictor to adapt to underlying system changes that might occur during a software update.

The AFP was validated using out-dated software and only worked under very specific circumstances. This research seeks to enable the validation of the AFP on modern systems. Specifically, the target of this research is a Microsoft Windows active directory domain services server. To that end, a full-stack authenticated session traffic is required in order to sufficiently load the service so that when failure is induced, it will happen quickly and in a realistic way. At the time of publication, we could find no such generator.

2.2 Network Load and Traffic Generation

While many tools for the purpose of generating network traffic exist, we could not find any that are capable of generating traffic with the intent of creating computational load for cryptographic systems such as the Microsoft Domain Services. In general, existing tools are classified into three categories: application, flow, and packet generators [3,4]. Application-level generators emulate traffic produced by applications on a network, flow-level generators replicate traffic using statistical modeling, and packet-level generators craft and inject packets into a network. Network traffic generators are further classified as open- or closed-loop. Open-loop generators use a packet arrival model for packet timing, whereas closed-loop generators wait for a response to a sent request prior to sending the next request [5].

At the time of publication, none of the tools available generate the necessary interaction with a deployed Microsoft Windows active directory environment necessary to facilitate the implementation of the AFP framework. Active directory implements the Kerberos authentication protocol in Windows domains and due to its cryptographic nature cannot be tested against replayed or random traffic; rather, a sequence of valid and invalid requests and responses are necessary to stress test this framework. Indeed, multi-step "handshakes" are necessary for rich service delivery and this capability is not realized by the current tools with any degree of modularity or extensibility.

A brief review of the traffic generators considered when researching this problem follows. The Distributed Internet Traffic Generator (D-ITG) [3] is, as its name implies, a distributed traffic generator capable of performing application, flow, and packet-level generation using both open- and closed-loop operations – sessions are initiated at specific time intervals and, within each session, new requests are not sent prior to receiving a response to the previous request. Sadly, D-ITG currently only supports TCP, UDP, ICMP, DNS, Telnet and VoIP which does not suit our needs.

NTG [4] is an application-level, distributed network traffic generator which is both open- and closed-loop. A key feature of NTG, as it relates to our problem, is that it interacts with existing network services. Unfortunately, it is only limited to web, mail, and multimedia servers/services, which is insufficient for our purposes.

Swing [6] is a flow-level, closed-loop traffic generator that observes live network traffic, extracts distributions from the traffic, and generates new traffic in a manner consistent with the observed traffic distributions. While this tool provides the ability to generate statistically-realistic traffic from generators to listeners across a link, the lack of both two-way traffic and interaction with existing services (specifically authentication services) does not satisfy the requirements for our problem.

A final tool worth mentioning, while not a network traffic generator, is Microsoft's Active Directory Performance Testing Tool (ADTest) [7]. Official Microsoft documentation is limited [8–11], and ADTest is designed to assess the ability of Microsoft 2003/2008/2012 Active Directory Lightweight Directory

Services (AD LDS) servers to add organization units and users, and make various changes to Active Directory to aid in developing requirements for an AD LDS deployment. It is important to note that Microsoft no longer supports this tool [9]. Also of importance, ADTest is not capable of testing other services that rely on active directory domain services for authentication (e.g. RDP, SMB, etc.), nor can it be extended to do so, and is therefore insufficient for the goals of this research.

In general, these tools are sufficient for generating traffic in a network, but they do not generate the full-stack, two-way authentication necessary to create significant computational workload for active directory domain services. The AFP requires the target system be placed under realistic computational load before injecting faults to capture the most realistic failure data possible [1,12]. A tool to generate this type of traffic did not previously exist.

3 Distributed PowerShell Load Generator (D-PLG)

This research introduces D-PLG, a tool for the generation of realistic network traffic in a Microsoft Windows domain for the purposes of software testing or computational workload generation. D-PLG can be classified as an application-level closed loop traffic generator and is a basic Windows PowerShell script that uses native PowerShell cmdlets for all of its functionality ensuring that the most realistic traffic possible is generated without overburdening the client machines used for generating load. D-PLG offers what has not seen in other traffic generation products or tools by making actual service requests and producing actual challenges and responses for Windows authentication protocols.

D-PLG is written in the Windows PowerShell environment which provides a tremendous amount of power and flexibility to generate traffic exactly as the actual services run by users would. As a result, D-PLG does require the use of client machines. However, this work shows that generating this type of realistic traffic is possible by utilizing only a small number of machines, or without producing a noticeable burden on in-use client machines.

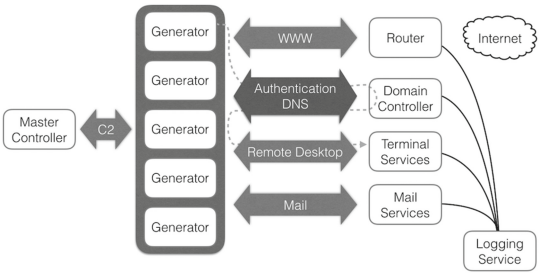


Fig. 1. How each type of traffic that is generated is routed. Log events are offloaded to logging service for further analysis.

The D-PLG architecture used for this work is shown in Fig. 1. D-PLG is most effective if used by a few client machines during idle downtimes but is developed in such a way that a user can still use a machine that is generating load, but may notice degraded performance depending upon how much traffic that particular client is being asked to generate. D-PLG is currently designed to be centrally controlled, asking a configurable list of clients to produce traffic for a fixed period of time.

D-PLG implements a feature that has not been previously seen and thus, a comparison with the existing tools is difficult. Relevant existing tools simply replay previously observed traffic. While this naïve approach may be highly representative of realistic traffic, it is not capable of creating any real computational workload for cryptographic systems. Modern cryptography relies on random and dynamic challenge-response protocols, as a result any inbound requests that are not capable of generating dynamic challenge responses are typically dropped immediately.

At the time of writing, D-PLG is comprised of three modules capable of generating full-stack web requests, Microsoft remote desktop protocol, Microsoft server message block (SMB) file sharing, and all associated authentication traffic. An intended byproduct of all of this traffic is domain name system (DNS) requests. An important part of any active directory domain is DNS and as a result, no load generator would be complete without performing DNS lookups.

The rest of this section outlines each of the three modules currently implemented as well as plans for future modules.

3.1 Web Browsing

D-PLG is capable of generating full-stack web requests and presently simulates an actual user browsing. This module is implemented using the 'Invoke-WebRequest' PowerShell cmdlet which upon completion returns an object representing the full document object model (DOM). With the DOM, D-PLG can programmatically simulate random browsing within a returned web page by completing and submitting web forms and requesting multiple different pages in a session. This functionality is different from the functionality implemented in many of the existing tools that only generate one-way transmission of the web request. As a result, this module allows D-PLG users to generate realistic load against web servers and potentially automate realistic web application testing.

3.2 Remote Desktop Protocol

Remote Desktop Protocol (RDP) is a simple protocol that allows the sharing and remote control of a Windows desktop environment. This module was included to generate authentication traffic with the active directory domain services server as well as place computational workload on the remote desktop services server. Applications for this module could include network infrastructure capacity and server sizing planning. The module takes advantage of a modified third party cmdlet [13] which invokes a call to the native windows remote desktop application

(mstsc.exe). A single modification was made to tell the cmdlet not to present a window as to avoid interrupting an individual who may be using the computer at the time of load generation.

Currently, the RDP module makes a full-stack remote desktop connection with an RDP server without producing a window which can allow us to take advantage of clients in active states. The script then sleeps for a few seconds and then closes the connection. Future versions will implement some sort of actual interaction with the RDP server like file upload or application use. This functionality was based on a tool previously developed by Microsoft [14] which is no longer maintained as evidenced here [15].

3.3 Server Message Block (SMB) and File Sharing

D-PLG implements an SMB file sharing module that connects to a local or remote share, creates a file in the share, fills that file with random ASCII data, saves the file, deletes the file, and finally deletes the share. This sequence of operations ensures that full-stack SMB file sharing requests are utilized and thus, causing the domain services server to authenticate the transaction and the file sharing services server to process the data being uploaded. This simple module could additionally be used to ensure a file server is live before beginning more complex operations.

Like the other modules, the SMB module was rapidly built due to the flexibility of this framework and implemented in only fourteen lines of code. Future versions will implement a variable amount of upload data or allow the user to select his or her own file. By allowing the user to upload a custom file, this module could be used to test application aware firewall rules to ensure certain types of files are or are not allowed to traverse a network.

3.4 Possible Future Modules

Many core active directory domain services have been implemented as a proof of concept, it should be noted however that implementing additional services is trivial. For example, simple message transfer protocol (SMTP) traffic could be implemented in a single line of PowerShell code using the 'Send-MailMessage' cmdlet. Additionally, the 'Out-Printer' cmdlet would allow for the sending of realistic full-stack network printer traffic. To facilitate future development, D-PLG is published in its current form under the MIT license on GitHub[1].

These modules demonstrate the extensibility of D-PLG and are capable of generating traffic that is representative of sophisticated network interactions that are necessary to create a performant workload generator for the tableau of modern networking services.

[1] https://github.com/paullj1/master/D-PLG.

4 Experimental Methodology

The following two subsections outline the experiments designed to validate D-PLG. The first describes in detail the virtual test environment. The following section, details the experiments which utilized the virtual environment.

4.1 Virtual Environment

The virtual environment was hosted on two VMWare ESXi 5.5 hypervisors each with two 2.6 GHz AMD Opteron 4180 (6 cores each) CPUs and 64 GB memory. The individual virtual machines are detailed in Tables 1, and 2. D-PLG uses cmdlets that did not exist until PowerShell version 3.0 so each of the Microsoft (MS) Windows computers had the MS Windows Management Framework version 4.5 installed. The installation of this framework also necessitated the installation of the MS .NET Framework version 4.5. In an enterprise environment these software frameworks would more than likely already be installed as they are part of the service pack updates that have since been released by Microsoft.

Table 1. Hypervisor 1.

Qty	Role	Operating system	CPU/Mem
1	DC	Win. server 2008	2/2 GB
5	Client	Win. 7	1/512 MB

Table 2. Hypervisor 2.

Qty	Role	Operating system	CPU/Mem
1	RDP	Win. server 2008	1/4 GB
1	Log	Ubuntu 14.04 LTS	1/1 GB

In addition to the requisite software being installed, each client was added to the domain and required a few minor modifications. First, D-PLG creates remote 'PSSessions' on each client machine and then invokes the cmdlets that have been assembled to generate the desired load. To enable these sessions, the credentials of the controller must be delegated to the clients so that they may be used to perform actions on behalf of the controller. This delegation is done very simply through the PowerShell cmdlet 'Enable-WSManCredSSP'. The final modification was for convenience; a copy of the scripts to be executed remotely was placed on the desktop of the Administrator user.

The domain controller had two MS Windows Server roles enabled: active directory domain services, and domain name service (DNS). One domain administrator account was used control the load generation, and individual user

accounts were created for RDP use and simple authentication traffic. The RDP server only had one MS Windows Server role enabled: remote desktop services.

The Ubuntu server was deployed and used as a central syslog repository for analyzing load on the domain controller and RDP server. The default rsyslog application was simply configured to accept incoming connections and then the rsyslog Windows agent was installed on the domain controller and RDP server.

D-PLG is organized into two scripts. The first is the 'LocalLoadGen' script and is placed on each client computer. It should be noted here that this practice may not be ideal in a production environment, but the placement and removal of this script could easily be automated when the controller runs. The second script 'RunLoadSim' is designed to act as a command and control element that connects to each client and executes the 'LocalLoadGen' script as an asynchronous job. For the purposes of this research, the command and control script was executed from our RDP server.

4.2 Experiment Design

Two experiments were designed to validate, test, and demonstrate the efficacy of D-PLG and are detailed here. In both of the following tests, D-PLG was run five times, where each execution consisted of five minutes of traffic generation within the virtual environment. The domain controller was sized based on the Microsoft community recommendation for up to fifteen thousand users in [16]. The goal of these experiments was to produce sufficient traffic to achieve the level of workload that a production domain controller should be able to sustain based on how its size. ESXi's reporting tools were used to measure success by collecting the relevant data in the form of packet captures at the virtual switchports of one client machine, the terminal server, and the domain controller. Further data collected came from the ESXi performance data. After each test, the performance data were exported from each of the hypervisors on the terminal server, one client, and the domain controller. In these data, CPU utilization, memory utilization, disk operations, and network traffic were reported on twenty second intervals. Finally, as previously stated, the rsyslog Windows Agent was used to forward log events from the domain controller and RDP server to an Ubuntu server. These log entries were then split into pieces that corresponded with each round of the tests.

The primary question we wanted to answer is, how much traffic can a PowerShell script really generate, and is it enough to sufficiently load an enterprise domain controller? The first experiment was designed to answer that question. To maximize the amount of traffic and subsequent workload generated, the client machines were only configured to make a single request. To do this, the 'RunLoadSim' was only tasked to perform a basic authentication request to the domain controller. The goal was to maximize the number of authentication requests the server could handle based on the way it was sized. In this case, that number was fifteen thousand users and a goal CPU utilization of 40%. To prevent overburdening the client machines, we found that the highest frequency at which these events could be created and handled was 10 per second. Fortunately,

five clients running for five minutes making ten requests per second equated to exactly fifteen thousand requests.

It should be noted here that the client machines used were significantly less powerful than average desktop computers typically found in an enterprise environment. Each authentication event took 20 ms so the maximum number of requests per second that was observed was fifty. As a result, this same experimental setup can sufficiently load a domain controller sized for seventy-five thousand users.

The second experiment was designed to determine how much load could be produced without having a significant effect on the resources available to each client machine. We configured the clients to utilize each of the modules that are currently implemented in D-PLG. In each round of the test the client machines looped continuously making an authentication request to the domain controller, a full RDP connection, an SMB share connection, a web request to a randomly selected URL, and finally a web request to a URL randomly selected from the page returned by the first request. The loop was configured to run twice per second, however due to the high latency of the web requests, the client was not expected to make that many requests every second of the test. This configuration choice was made to ensure maximum utilization when possible.

5 Experimental Results

In this section the results and data collected after conducting the tests described in Sect. 4.2 are detailed. To answer the quantity question, the number of packets produced per second was explored as well as CPU utilization, memory utilization, log events, and network operations on the domain controller. In this first round of tests, the domain controller reported an average of 56,291 log events over each five minute test or approximately 187 log events per second. In addition, an average of 6,267 packets per second were captured over each of the five tests. Figure 2 shows the distribution on the number of packets sent and received by the domain controller for each test and tells us that the load was consistently high throughout each test. On the client side, as predicted, the load was also relatively high as seen in Fig. 4.

The load generated against the domain controller was consistently at 40% which is exactly in-line with the amount of load it should be expected to endure during peak business hours per the Microsoft community recommendations for sizing [16]. Unfortunately in this case, the client machines would likely not have been usable during the test. Fortunately, because only five low-end are needed machines to produce this load over a relatively short period of time, a simple solution to this problem would be to purchase five inexpensive desktop computers for this purpose, or conduct testing during an idle downtime (Figs. 3 and 5).

The second test evaluates whether the client machines could produce a sufficient amount of realistic traffic without being over burdened so that they could be used to generate load even as individuals use them. To answer this question, CPU utilization, memory utilization, and packets transmitted per second were

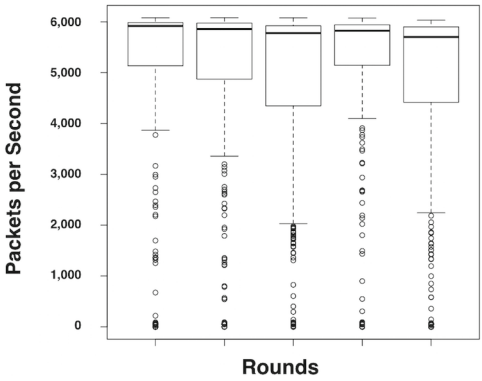


Fig. 2. How many packets per second were sent or received by the domain controller across all five rounds of the first test. In each test, approximately 1.8 million packets were captured.

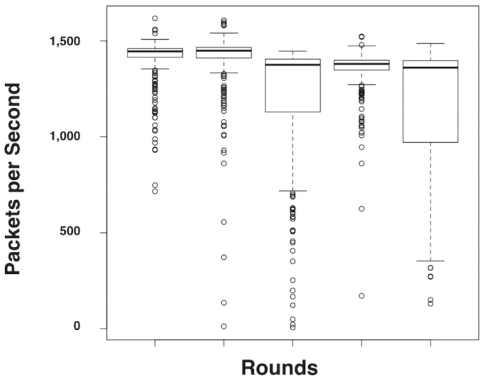


Fig. 3. How many packets per second were sent or received by one of the clients across all five rounds of the first test.

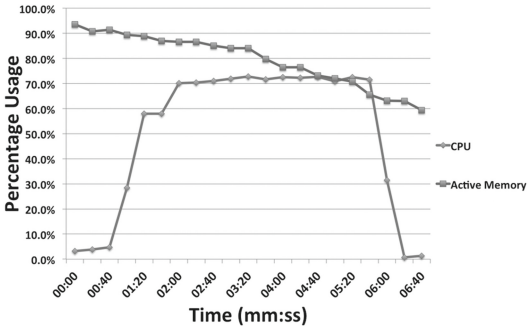


Fig. 4. Client CPU and memory utilization during the first test.

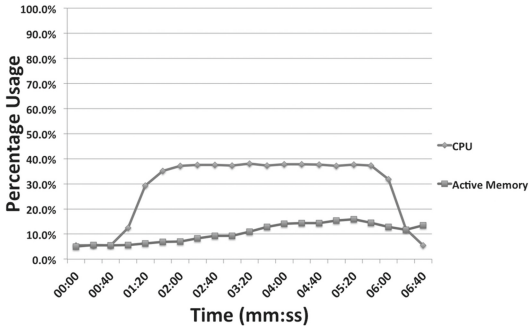


Fig. 5. Domain controller CPU and memory utilization during the first test.

examined with respect to the client. The average number of packets generated over the five minute tests was 5,499 and the remainder of the data can be seen in Fig. 6. These same data with respect to the domain controller and RDP server were examined as shown in Figs. 7, and 8 respectively.

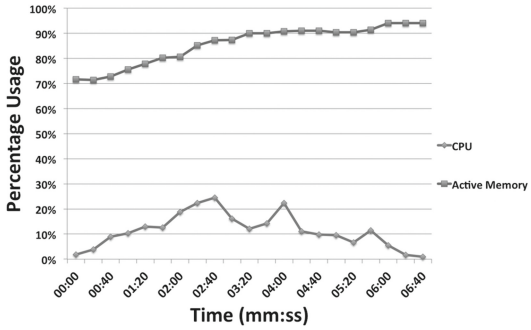


Fig. 6. Client CPU and memory utilization during the second test.

While the number of packets sent was relatively high, these results do not demonstrate a sufficient amount of load on either the domain controller or RDP server. We suspect that this is due to the majority of the time spent during the test retrieving web-pages. As a result, if a proxy server or application aware firewall is the target of this load generation, this level of traffic is sufficient for that purpose. These results also show that a client computer asked to generate traffic could still be used during a test. In future work, this infrastructure could be tested with non-blocking web-requests so that more load can be placed against the local services while web requests are being processed externally. Alternatively, more client machines could be used during the test.

In general, the results observed lead us to believe that D-PLG an be extended and used under any circumstance where dynamic network transactions are

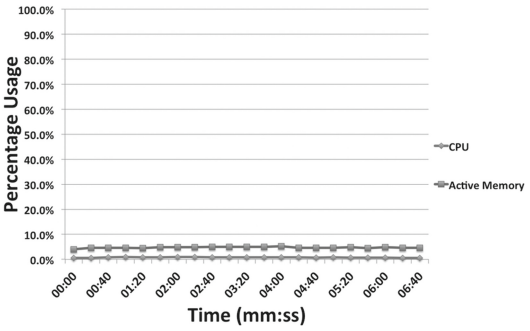


Fig. 7. Domain controller CPU and memory utilization during the second test.

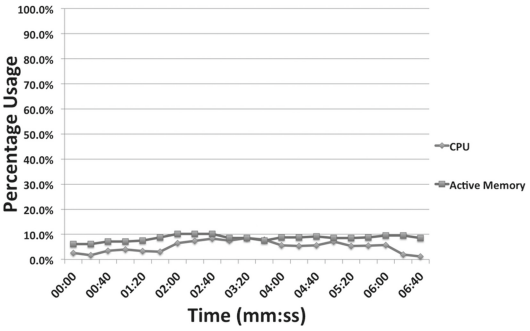


Fig. 8. RDP server CPU and memory utilization during the second test.

required between unbounded network components. Further, these results show that D-PLG can be leveraged by the AFP framework in future work.

6 Future Work

There are many opportunities for improvement in D-PLG. If D-PLG is to be used to simulate realistic network traffic patterns that would normally be generated by humans, much work would have to be done to balance the kinds of requests that get made. For example, a typical user might log in, browse the web for a few minutes, check his or her e-mail, then maybe send an e-mail. Currently, D-PLG is extremely predictable with respect to what kind of request it will make next. The framework can be made a lot more relevant with programmatic generation schemes such as REGEX-based pattern generation or training of input validity via machine learning. For the purposes of implementing the AFP however, the level and quality of load observed is sufficient.

More configuration options could be added like the depth of a browser simulation or having the browser simulate filling out web forms using configurable

data. D-PLG could also allow for finer grain control over the SMB module allowing users to select a file or specify the size of the randomly generated file. Most of these configuration options would be implemented in a straightforward way.

Finally, as previously discussed, other modules which take advantage of more of the native Windows PowerShell cmdlets like 'Send-MailMessage' and 'Output-Printer' could be implemented with relative ease.

7 Conclusion

Based on the results of the tests presented in this work, D-PLG is capable of providing sufficient load of network services in a Microsoft Windows enterprise domain. Five clients were able to generate fifteen thousand authentication requests over a five minute time period which was very near the limit that the domain controller should be expected to handle based on the Microsoft community recommendations. Further, since the configuration was based on these recommendations, these results should generalize and scale for larger networks. The use of client machines to produce this load is negligible and have proven that it can be done centrally, with only a few machines, without placing a significant burden on those client machines, and without installing any additional software by use of the native Windows PowerShell cmdlets. Finally, if necessary and client machines are used during idle down times, significant load can be generated by D-PLG.

As a result of this work, D-PLG was successfully implemented and used to experimentally validate additional fault loads to extend and generalize the AFP framework [17]. This generalization allows for the implementation of the framework on modern Microsoft systems that is able to predict a wider range of faults than before.

Finally, there are many established needs for having network traffic and load generators. This work has demonstrated one important role D-PLG can play, and results suggest that D-PLG can fill many other needs for dynamic traffic generation. For example, in cybersecurity training events, traffic generators are used to simulate real traffic to mask malicious traffic. Other uses include equipment sizing, stress testing, and software testing. D-PLG can fill these needs as well and in general can be naturally extended and used under any circumstance where dynamic network transactions are required between unbounded network components.

Acknowledgements. This work was supported by the U.S. National Security Agency, National Information Assurance Education and Training Program (Alice Shafer and Glenn Ellisonn, Program Managers).

References

1. Irrera, I., Vieira, M., Duraes, J.: Adaptive failure prediction for computer systems: a framework and a case study. In: Proceedings of the 2015 IEEE 16th International Symposium on High Assurance Systems Engineering (HASE 2015), pp. 142–149 (2015)
2. Salfner, F., Lenk, M., Malek, M.: A survey of online failure prediction methods. ACM Comput. Surv. (CSUR) **42**(3) (2010). Article No. 10
3. Botta, A., Dainotti, A., Pescapé, A.: A tool for the generation of realistic network workload for emerging networking scenarios. Comput. Netw. **56**, 3531–3547 (2012)
4. Zach, P., Pokorny, M., Motycka, A.: Design of software network traffic generator. In: Recent Advances in Circuits Systems, Telecommunications and Control, pp. 244–251 (2013)
5. Weigle, M., Adurthi, P., Hernandez-Campos, F., Jeffay, K., Smith, F.: Tmix: a tool for generating realistic TCP application workloads in ns-2. ACM SIGCOMM Comput. Commun. Rev. **36**, 67–76 (2006)
6. Vishwanath, K., Vahdat, A.: Swing: realistic and responsive network traffic generation. IEEE/ACM Trans. Netw. **17**, 712–725 (2009)
7. Microsoft: Download Active Directory Performance Testing Tool (ADTest.exe) from Official Microsoft Download Center (2012)
8. Bijaoui, P.: Microsoft Exchange Server 2003 Scalability with SP1 and SP2. HP Technologies. Elsevier Science, Boston (2011)
9. Morowczynski, M.: How To Use the Active Directory Performance Testing Tool on Windows Server 2012—Ask Premier Field Engineering (PFE) Platforms (2014)
10. Suyanto, H., Tiwari, M.: Windows 2008 AD LDS Load Testing using ADTEST - Part 1 - TechNet Articles (2010)
11. Suyanto, H., Tiwari, M.: Windows 2008 AD LDS Load Testing using ADTEST - Part 2 - TechNet Articles (2010)
12. Irrera, I., Vieira, M.: A practical approach for generating failure data for assessing and comparing failure prediction algorithms. In: Proceedings of the 2014 IEEE 20th Pacific Rim International Symposium on Dependable Computing (PRDC 2014), pp. 86–95 (2014)
13. Brasser, J.: Script connect-mstsc - open RDP session with credentials (2015)
14. Microsoft: Download remote desktop load simulation tools from official microsoft download center (2009)
15. Szeto, A.: Debug Assertion Failed: sockcore.cpp, line 623 (2012)
16. Makbulolu, S., Geelen, G.: Capacity planning for active directory domain services. Technical report, Microsoft Corp (2012)
17. Jordan, P., Peterson, G., Lin, A., Mendenhall, M., Sellers, A.: Data driven device failure prediction. Master's thesis, Air Force Institute of Technology, Defense Technical Information Center (DTIC) (2016)

HLogo: A Haskell STM-Based Parallel Variant of NetLogo

Nikolaos Bezirgiannis[1(✉)], I.S.W.B. Prasetya[2], and Ilias Sakellariou[3]

[1] Centrum Wiskunde & Informatica (CWI), P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
n.bezirgiannis@cwi.nl
[2] Department of Information and Computing Sciences, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
s.w.b.prasetya@uu.nl
[3] Department of Applied Informatics, University of Macedonia, 156 Egnatia Street, 54636 Thessaloniki, Greece
iliass@uom.edu.gr

Abstract. Agent-based Modeling and Simulation (ABMS) has become a quite popular approach among researchers in the community, mainly due to its simplicity, expressiveness and wide applicability. However, in most cases, ABMS tools demonstrate reduced performance, especially when dealing with large experiments. This paper presents HLogo, a parallel variant of the NetLogo ABMS framework, that aims to increase the performance of simulations by utilizing Software Transactional Memory and multi-core CPUs, while maintaining the user friendliness of NetLogo. HLogo is implemented as a Domain Specific Language embedded in the functional language Haskell, which means that it also inherits Haskell's features, such as strong static typing, a module system and a vast collection of programming libraries.

Keywords: Agent-based Modeling · Agent-based simulation · Concurrent agent-based simulation · Concurrent NetLogo.

1 Introduction

Simulating a system as a set of interacting agents has gained significant popularity in the past decades. The approach proved to be both natural and widely applicable in various areas, giving rise to the so called Agent Based Modelling and Simulation (ABMS) field. In the latter, complex system behaviour often emerges from the interaction of relatively simple agents and simulation allows to study these emergent phenomena. ABMS has been shown to have broad applicability, e.g. in social sciences [13], ecology [15], biology [28], and physics [39]. As a consequence, numerous ABMS frameworks have been proposed [6,29,36], and research has focused on various aspects of the latter, such as methodology [32],

M.S. Obaidat et al. (eds.), *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing 676,
https://doi.org/10.1007/978-3-319-69832-8_7

ease of use [40], portability [2], and expressiveness [31]. The size of agent populations may be a crucial factor towards the emergence of certain sought-after system phenomena, so performance is also an important aspect. The ability to scale up simulations to large populations comes down to, basically, how much computation the used ABMS framework can "crunch" per time unit. However, till recently little attention was devoted in improving the performance of available ABMS frameworks [30].

This paper extends our previous work reported in [4] by introducing in greater depth a new ABMS framework called HLogo. The framework is strongly inspired by the well-known NetLogo framework [38]. Like NetLogo, HLogo also strives for simplicity. For this reason HLogo is implemented, actually embedded, in a powerful and pure functional programming language called Haskell [1], hence the name "HLogo". Unlike NetLogo, HLogo is a *concurrent* ABMS framework, that aims to speed up simulations by harvesting the parallelism available in modern multi-core CPUs. HLogo offers three unique features which also constitute the main contribution of this paper:

- HLogo allows agents to run *concurrently*, with the latter implemented by utilizing a technology called Software Transactional Memory (STM) [33]. This allows the complexity of synchronizing agents to be completely hidden from the programmers, hence keeping HLogo just as simple as NetLogo, despite the concurrency. Coupled with Haskell's lightweight (green) threads, the overall ABMS execution enjoys significant benefits from *multi-core* parallelism.
- HLogo is *embedded* as a Domain Specific Language (eDSL). As a DSL it has a simple base syntax. As an embedded DSL it also inherits all the advantages of its host language, Haskell. For example, it inherits Haskell's module system (which NetLogo lacks), allowing HLogo programmers to import and use a plethora of Haskell libraries. The decision to embed the DSL, rather than directly implementing it, does mean however that its syntax is limited by that of its host language.
- HLogo is statically typed and it inherits Haskell's *type inference*. This strengthens the safety of HLogo programs without burdening the user with writing type annotations.

The rest of this paper is organized as follows. Section 2 outlines the NetLogo approach to ABMS, in order to explain the concepts implemented in HLogo. Section 3 presents related work in the area of Logo-like simulation platforms and parallel approaches to simulation. Some important features of Haskell, the implementation language of choice, are presented in Sect. 4. The HLogo language is presented in Sect. 5 with Sect. 6 discussing the parallel features of the former, and Sect. 7 reporting an experimental evaluation of the former. Finally, Sect. 8 concludes the paper and presents future research directions.

2 Agent Based Modelling and Simulation in NetLogo

The ABMS community has enjoyed the introduction of a significant number of simulation platforms that differ in a number of characteristics, such as modelling

approach, efficiency, user-friendliness, etc. NetLogo [38] is one of the most well known and widely used platforms, mainly due to its simplicity, small learning curve and the ease by which users create *simulations*, or *experiments* as the former are referred to in NetLogo.

Three kinds of agents exist in the NetLogo environment, namely *patches*, *turtles* and *links*, each modelling a different aspect of the simulation world. All entities are stateful and active, i.e. they have a set of attributes (variables) which determine their state and the user can encode agent behaviour using the provided domain specific language. The latter offers a set of primitives that target encoding of both agent perception and action mechanisms, simplifying the task significantly.

Patches are stationary agents, that form the two dimensional (or three dimensional) world, i.e. the grid on which turtles "live". The dimensions of the grid are fixed during the execution of an experiment and in the case of patches, variables and code allow modelling of complex environments. Turtles are agents that are dynamically created during the experiment and can move on the grid with links connecting two turtles, i.e. representing a relation between the latter. Agents can be organized into breeds, although the depth of such organisation is limited to one. Each breed can have its own user-defined attributes, apart from the system-specified ones. For instance, this declaration:

```
breed [cows cow]
cows-own [age hunger]
```

defines a new breed called `cows`, with each of its member having attributes `age` and `hunger`.

One of the important notions in programming NetLogo agents is that of an *agentset*, i.e. a set of agents of a specific type (either turtles, patches or links) that have certain characteristics. For instance, all agents of a specific breed are part of the agentset with the same name (e.g. `cows`). More interesting agentsets are formed with the use of NetLogo primitives, as for example `cows in-radius 3` which forms an agentset of all cows located around the calling agent at a specific distance. A rich set of primitives similar to the above allow implementation of agent perception mechanisms. Among those are the `with` primitive that defines a boolean condition on the agentset formation, i.e. the line `cows with [age>4]`, will collect all cows older than 4 years old.

Execution involves *asking* an agent or a set of agents to perform some action. For instance, the following line of code:

```
ask cows in-radius 3 [set hunger 0]
```

commands all cows inside the specific radius to set their hunger attribute to 0. The *observer* entity is the initiator of the experiment, and can ask other agents to execute some code, and every agent can ask other agents to perform some action. The built-in variables `self` and `myself` refer to the currently executing agent (similar to 'this' in OO), and the parent caller that `asked` this agent, respectively. Besides built-in commands an agent can execute custom commands

defined by user-defined *procedures* and user-defined functions called *reporters*, in NetLogo terminology.

Finally, the state of other turtles can be queried by using the "of" primitive. For example, the expression [age] of cows reports back a list of all cows' ages in the simulation world. In addition to reporting an attribute, "of" can also evaluate a function on the target agent's context and report the result.

The approach, briefly outlined above, has become widely accepted by researchers using ABMS in a number of fields, with the list of publications citing NetLogo becoming quite large and increasing steadily each year, proving that the environment has sufficient modelling capabilities. However, building large experiments is currently not sufficiently supported, mainly due to the fact that the execution model is sequential. During an ask command as the one shown above, each agent in the set has to complete the execution of the code given to it, before the next agent in the set can start. Although this approach does solve a number of problems, it simply cannot take advantage of the computational power offered by current multi-core processors.

Running a simulation in parallel is a rather complex task, since executing agents have constant access to a shared environment as well as each other's states. Handling concurrent changes creates a major challenge that any parallel simulation platform must address. This work addresses the problem by introducing a NetLogo variant implemented in Haskell, relying to the Software Transactional Memory control mechanism provided by the later.

3 Related Work

Since this work deals with a parallel ABMS platform developed in Haskell, this section first discusses existing implementations of NetLogo ABMS platforms, then parallel simulation platforms, and finally existing simulation frameworks in Haskell.

NetLogo [38] models are written in a dialect of the educational programming language Logo. The language is dynamically typed with lexical variable scoping and is implemented in the Scala programming language. A compiler translates NetLogo code to Java bytecode, to be later run in a Java Virtual Machine (JVM). The platform includes a GUI to visualize simulation results, and a rich collection of predefined models. A limited form of type checking based on agent types (turtles, patches etc.) is supported, i.e. there are certain commands that can only be run in a specific agent context and a user defined procedure that contains them must be run in the same context as well.

ReLogo [24] is a NetLogo clone embedded as a DSL in Groovy (an OO language running in JVM) that comes as a part of the Repast simulation suite. As is the case with NetLogo, ReLogo is single-threaded and comes with a rich GUI & IDE based on Eclipse. Although Groovy 2.0 introduced optional static typing, ReLogo cannot type-check many of its expressions: agentsets are untyped, and ask/of/with closures cannot track the type information of their context (self, myself). The simulation user has to either resort to type-casting or turn off Groovy's static typing in the pertinent code.

In order to speedup large sets of experiments, both NetLogo and ReLogo support *parameter sweeping* [19]: running *multiple instances* of the same model in parallel, each on its own CPU core, while varying the model's input parameters. Taking a more traditional approach, HLogo, tries to inject parallelism *inside a single instance* of a simulation run; this can be crucial for large models or time-critical simulations where any performance gain is desirable.

To the best of our knowledge, the work described in this paper is the first to apply Software Transactional Memory in Agent-Based Modelling. However, there are other approaches in the literature that investigate the issue of speeding up ABM execution using various parallel techniques: the work described in [21] proposes to execute Agent-based systems through Distributed Discrete-Event Simulation. The key problem as reported in the paper, is the decomposition of the environment which leads to the problem of fair load balancing of the distributed machines. SPADES [30] is another Distributed Agent Simulation Environment that explicitly models the full agent cycle (sense-think-act), while having distributed execution and reproducibility of results. The work reported in [22] employs a well known parallelization technology, OpenMP, to speed up the execution of agent-based models. However, the technique restricts the implementation language of ABM frameworks to only those which provide an OpenMP implementation, i.e. C, C++, Fortran. It also adds the burden of annotating simulation code with extra OpenMP pragmas, which is rather discouraging for simulation developers. SASSY [16] is a scalable agent based simulation system that acts as a middle-ware between an agent-based API and a Parallel Discrete Event simulation (PDES) kernel. The difference in SASSY compared to [21,30] is that the ABM framework can be built up from existing standard PDES kernels. An innovative method of executing mega-scale Agent-Based Models in the massively parallel Graphics Processing Unit (GPU) is proposed in [10]. Although, it is well established that this method can lead up to considerable speed gains, we feel that the expressiveness of Agent-based models that can be run on this platform is rather restricted. A similar framework is Flame GPU, built on-top the FLAME ABM framework [17], and has successfully been applied on project EURACE to simulate the European economy model [8].

Within the Haskell community, our work is the first Logo-based simulation framework implemented in Haskell. Other simulation packages for Haskell are for example *Hasim* [3] and the recently introduced *Aivika* [35]; both are libraries for Discrete Event Simulation (DES). *Hasim* [3] provides process-based DES, however it does not employ any kind of parallelism. *Aivika* provides DES with extensive system dynamics. There is also the *event-monad* library that provides a monad (see also Sect. 4.2) and monad transformer for events; it can be used as a low-level helper library to build a simulation framework. Users can create an event-graph simulation system and schedule events to it. In principle, it does not employ any parallelism, but it could theoretically be used together with some parallel strategy to exploit parallelism.

4 Haskell

The functional programming language Haskell [1] was selected as the HLogo implementation platform, for two main reasons. Firstly, it is an excellent choice for embedding domain specific languages (DSLs) [14] and HLogo is designed as a DSL. Secondly, Haskell offers an excellent implementation of Software Transactional Memory (STM) [9], which is crucial for realizing HLogo's parallelism. This section will introduce several concepts from functional programming necessary to explain the embedding of HLogo in Haskell, and introduce STM.

4.1 Static Typing

An important feature that HLogo gets from its implementation in Haskell is that the latter is a statically typed language. This extends to HLogo as well, which is in contrast to NetLogo's dynamic typing. For example, in Haskell, and thus also in HLogo, a type error in an expression such as `1 + non_number` will be detected at compile time. It should be noted that NetLogo also checks type consistency of its expressions, but the majority of such checks is done at runtime and consequently, such errors, as in the example above, are detected rather late. In this respect, HLogo can be said to provide more safety for agent-based modeling. However, if the user needs dynamic typing, e.g. for its flexibility, it can be supported in Haskell through the Data.Dynamic module.

HLogo also takes advantage of Haskell's powerful type system and thus can type-check not just simple arithmetic expressions, but also more elaborate statements. The majority of built-in Logo-like commands need to be executed in a specific agent context, e.g. the command `forward` n may only be executed by a turtle agent (other types of agents are immovable). The example below presents a NetLogo expression, and its HLogo counterpart, where we erroneously "ask" the patch at location (0, 0) to die:

```
%% NetLogo version: yields error only later at runtime
ask patch 0 0 [die]

-- HLogo version: this program does not type-check
ask (atomic die) =≪ patch 0 0
```

In this case, HLogo will detect the error successfully at compile time. Elaborating more on this error, the action `die` should only be invoked on either a turtle or a link, whereas patches are to live through out the simulation, so they should not "die". In Haskell, this is enforced by overloading the name `die`, which is achieved by defining `die` as an operation of a 'type-class' called `TurtleLink`. Haskell *type-classes* offer a similar concept to interfaces in OO languages, e.g. Java, used for ad-hoc polymorphism. A type-class defines a set of operations, but it only defines their signatures, and thus provides no implementation. When the type T is declared to be an instance a type-class, e.g. `TurtleLink` (in OO jargon we would say T 'implements' `TurtleLink`), T gets all `TurtleLink`'s operations including `die`, but on the other hand the declaration must specify how

T implements those operations. In HLogo turtles and links are declared to be instances of `TurtleLink`, thus allowing `die` to be overloaded for both types of agents. On the other hand, patches are not instances of the former type-class; therefore the action `die` in the above code yields a type error.

Haskell's strong type system also comes with Hindley-Milner type inference [23], which makes type annotations optional. This provides type safety to the user, without the burden of annotating the code with type signatures. For example, the following command is a variation of the previous example:

```
ask (atomic (do {forward 3 ; die})) =≪ agentT
```

There is no need to explicitly annotate the type of `agentT`. Haskell type system can infer that it must be a turtle: it "knows" that `die` expects an agent which is an instance of `TurtleLink`, whereas forward expects a turtle (patches and links are not supposed to move around). So by implication Haskell successfully infers that the agent on which the above commands are applied has to be of type Turtle.

4.2 Monads

In Haskell, types can be parameterized. For example, lists of integers are of type [`Int`]. Actions that perform input/output (IO) are instances of the type `IO` a, where a is the type of the result of such an action. For instance, the function `getChar` that reads from the console and returns the read character has the type `IO Char`. The [.] and `IO` parts in these examples are called *type constructors*; they construct a new type from the type given to them.

Being a purely functional language, Haskell has no natural concept of side effect and thus modeling agents that are stateful and have actions with side effects on their own or other's states and the environment does not come naturally. However, such states and side effects are brought into the language through monads [27]. In functional programming, a *monad* is a generic concept for composing items through an associative operator. A simple example of a monad is lists with the concatenation operator. It should be noted that in Haskell function compositions are allowed and thus monads can be used to compose computations. In more technical terms, a Haskell `Monad` is a type-class and for the purpose of this discussion, we will assume that this type-class offers an associative operator ";" for composing monadic actions[1]. A type constructor `M` can be made an instance of `Monad` by providing a concrete definition for ";". If `M` is a monad, then an expression of type `M` a is a monadic action: when executed, at the end will produce a value of type a. For example, the previously mentioned type constructors, `IO` and [.] are both monads and the function `getChar` is thus a monadic action. The set of commands for HLogo agents, such as `forward` and `die` also form monads.

[1] The actual definition of `Monad` offers a slightly different and richer set of operations [26].

Monadic actions can be *sequentially composed* with the **do** notation. E.g., suppose c is a monadic action of type `M Int`, then the expression: **do**{`z` ← `c`; **return** (`z`+`1`) } is a new monadic action[2]. It first evaluates/executes c, then binds the produced value in the variable z, then $z + 1$ is returned as the produced value of the whole action. The concept is general through the overloading of ";" operator (recall that `Monad` is a type-class) and the exact meaning of the example **do** expression above depends on the used monad (which `M` is being used). If it was the IO monad, or the monad of agents' commands, the meaning is roughly as described above. If the monad was the list monad, it would have the net effect of constructing the list $[\, z + 1 \mid z \in c \,]$.

The expression **do**$\{e_1\ ;\ e_2\ ;\ ...\}$ will evaluate the expressions e_1, e_2, ... in the given order. When e_1, e_2, ... are written vertically, and start at the same column, we can drop the use of delimiters "{","}", and the ";" to get a cleaner syntax. A **do**-sequence that does not explicitly specify a **return** as the last expression, implicitly returns whatever the last expression returns. For example, the HLogo expression below is a monadic action that first obtain the set of all patches in the simulation, and then it returns the number of elements of this set.

```
do p ← patches
   count p
```

There is also the $e =\!\ll d$ operator that we saw before in Sect. 4.1, that pipes the value returned by the monadic action d as an input for e. So, the above code can also be written more succinctly as `count` $=\!\ll$ `patches`.

4.3 Software Transactional Memory (STM)

STM is a concurrency control mechanism that allows concurrent processes to be programmed without having to synchronize them, and yet allowing them to safely operate on common states. This makes the task of programming such processes much easier and much less error prone. There is no need to worry about race conditions, nor deadlocks. STM's concurrency relies on the so-called *transactions*—a concept that originally comes from the database domain. A transaction is a sequence of reads and writes to a set of so-called *transactional variables* or *TVars*. A TVar represents an actual store in the memory, which can be shared by multiple transactions. The execution of a transaction is *virtually atomic*, that is, its intermediate changes on the TVars it operates on cannot be witnessed by any other transaction. The idea originates from the field of Distributed Databases, where a transaction corresponds to an atomic SQL query or update. Whereas database transactions operates on tables' rows or columns, transactional memory operates at the level of memory stores. Historically, transactional memory was introduced in 80's by Knight as an extension to Lisp with suitable hardware modifications to enable concurrency [18]. In 90's, Shavit and Touitou turned the idea to a pure software implementation of the approach, and coined the

[2] The actual symbol in Haskell is <- instead of ←. We use the later just for improving the presentation.

term Software Transactional Memory. Recently, STM programs can be further accelerated through hardware instruction-set extensions, e.g. with Transactional Synchronization Extensions (TSX) of the Intel® Skylake processor.

Multiple transactions can run in parallel, however each transaction τ does not directly write to its TVars. Instead, it keeps a separate log of reads and writes to the TVars, with writes not committed yet. At the end of the transaction, it checks if one of its TVars has been modified with respect to its value at the start of τ. If no modifications have occurred, all τ's writes are committed and the transaction is said to be successful. Otherwise, τ is aborted, and later retried again.

Haskell has an excellent STM library [9] and furthermore, its strong type system guarantees that transactions are 'safe'. It is important that aborted transactions *have no irreversible side effects*, in other words, if they do have effects, we should be able to rollback those effects. This can present a problem, as for example in cases where within a transaction we perform IO actions (e.g. to read a value from the keyboard), which typically cannot be rollbacked. In most other language implementations, this cannot be enforced. In Haskell however, STM transactions and IO actions are both monadic actions, but of different types: a transaction that returns an integer will have the type `STM Int` whereas an IO action that read an integer from the keyboard has the type `IO Int`. Haskell type system guarantees that expressions of these monads cannot be intermixed, in the same way that Haskell does not allow an instance of `Int` to be subtracted from an instance of `Bool`.

In HLogo, a model typically consists of many agents. To gain parallelism, agents' actions can be composed from transactions. Internally, committing a transaction involves complicated orchestration where different places in the memory must be locked and unlocked in a certain order. However, this process is hidden from the programmers: from their perspective transactions are lock free, thus making concurrent programming much easier for them. For HLogo this is important, since we want to maintain NetLogo's user friendliness. Using STM does have its price. It introduces overhead due to aborted transactions. But still, experiences reported with STM in the literature suggests that considerable speedup can still be expected [25].

Ultimately, transactions are run by threads for concurrent execution. There are several options on how to do this. The obvious one is to run each transaction on its own thread. Haskell provides so-called green threads, i.e. virtual threads managed by a virtual machine (or by a language's runtime-system), as opposed to OS' native threads. Green threads use less memory and can be activated and synchronized faster. A large number (thousands; even millions) of green threads can be created without running out of memory. Haskell runtime-system employs an M:N threading model, where M green threads are automatically mapped to N OS (heavy weight) threads for multi-core parallelism. While this maximizes concurrency, in our case most of the time the number of available CPU cores is much less than the number of agents. The above solution would lead to performance degradation. Section 6 will discuss our solution to this.

5 HLogo

Both NetLogo and HLogo are instances of Domain Specific Languages (DSLs) for describing and simulating dynamic systems. A DSL is a programming language that offers, through appropriate notations and abstractions, expressive power focused on a particular problem domain [37]. NetLogo is a *native* DSL, i.e. it has a dedicated parser and a compiler or interpreter to execute its code, whereas HLogo is an *embedded* DSL (eDSL). An eDSL is embedded inside another language (host), usually a general purpose programming language and tries to *imitate* a native DSL by providing its language constructs in terms of the constructs of the host. It is an imitation in the sense that it may not look or even work entirely the same as the DSL it imitates, but it tries to. On the other hand, an eDSL can be more rapidly developed because we do not need a separate parser and compiler for it. An eDSL also inherits all the important features of its host language. HLogo inherits, among other features, Haskell's:

- expressiveness and its powerful type checking, as discussed in Sect. 4.1,
- module system, thus allowing us to organize HLogo agents into separate modules, a feature that NetLogo currently (as of version 5.3.1) lacks, and
- the already vast collection of open-source Haskell libraries (Hackage).

Haskell was chosen since it is a brilliant host language for embedding DSL's [14], as has been demonstrated in various cases [5,11,26]. In particular, the expressiveness provided by higher-order functions and type classes is crucial for imitating native DSL constructs. This approach also makes HLogo more easily extendible: new constructs can be simply added by the inclusion of more higher order functions, whereas in a native DSL we would need to modify the parser and interpreter.

It is true though, that as an eDSL the syntax of HLogo will be limited by the syntax of its host language, Haskell. For example, in NetLogo binary operators have higher precedence than function application; e.g. we can write: `print 1+3`. This is not possible in HLogo, because in Haskell the precedence is reversed. So, the same code in HLogo has to be written as: `print (1+3)`. As mitigation, Haskell's clean syntax and support for overloading can often be exploited to provide acceptable syntactical imitations of the original NetLogo constructs.

Nearly the complete set of NetLogo's standard library has been ported to HLogo. In the sections that follow, we limit ourselves to explaining how the main concepts are represented in HLogo. A complete example of a simple model is also included.

5.1 HLogo Agents

By default in each simulation there are always turtles, patches, and links. Identifiers of the corresponding names can be used to refer to them, e.g. `turtles` represents the set of all turtles in the model at hand. The dimension of the simulation world is set through command line and this determines the number of

patches in the model. Turtles can be created dynamically e.g. by the command `create_turtles` N. The command `ask create_link_with` β $=\!\!\ll$ α creates a link between α and β. These agents come with a number of pre-defined properties. E.g. turtles and patches have x and y coordinates specifying their position.

We can introduce a new breed of turtles and define new attributes for them. Technically, this requires the user to define a new Haskell datatype representing the breed, along with the corresponding set and get functions to access its attributes. To avoid having to write such boilerplate code, we employ Template Haskell [34], a compile-time meta-programming technique that will generate the needed code. As an example, the following code is part of the preamble of the example HLogo model in Fig. 1:

```
-- HLogo eDSL is a library
import Language.Logo

-- generates: cows,cows_here,...
breeds ["cows", "cow"]

-- generates getter/setter: energy
breeds_own "cows" ["energy"]
```

breeds and breeds_own are actually Template Haskell macros. The first creates, among other things, Haskell identifiers named `cows` and `cow` which can be used to refer to all cows, or to a specific cow (e.g. as in `cow 0 0`, the cow at position 0, 0). The second creates a new attribute for cows, which can be referred to by the identifier `energy`.

5.2 Commands

As mentioned in Sect. 2, NetLogo's main primitives for invoking commands on agents are: ask, of, and with. HLogo also provides these primitives (since "of" is a keyword in Haskell, the name "of_" is used instead).

The general syntax of ask is ask c $=\!\!\ll$ α, where c is the command to invoke and α is an agent (or an agentset) and which has the obvious effect of invoking c on α (or all members of α). More precisely, a *command* like c is a monadic action. The monad has a quite rich structure, but abstractly we can view commands as instances of the `IO` monad. For example `xcor` and `ycor` are commands that return respectively the x and y coordinates of the given agent. Other examples include `forward` and `die` mentioned in Sect. 4.1. These are actually STM transactions, which can be turned into commands by wrapping them with the function `atomic`—the connection will be discussed later. The whole construct ask c $=\!\!\ll$ α is again a command; it simply returns void.

Since these are monadic actions, commands can be composed with the **do** notation (see also Sect. 4.2), e.g. **do** $\{c_1; ...; c_n\}$, and we can invoke them on a set of agents, e.g. all turtles, as in:

ask (**do** $\{c_1; ...; c_n\}$) $=\!\!\ll$ `turtles`

The primitives of_ and with can be used with similar syntax: of_ c =≪ α and with c =≪ α. The primitive of_ is actually just a slight variation of ask; whereas ask always returns void, of_ c =≪ α returns whatever value c returns. If α is an agentset, the construct then returns a list of the results of invoking c on every member of α. The primitive with expects c to be of type `IO Bool` and α to be an agentset. It returns a new agentset consisting of those members of α, on which c returns true.

5.3 Procedures

NetLogo allows procedures to be defined through the to ... end syntax, for instance the code below defines the move[p] procedure, which will turn all cows 5° to the right, then move them p points forward:

```
to move[p]
   ask cows [right 5  forward p]
end
```

In HLogo, the same definition is achieved by a top-level function bound to its corresponding right-hand side monadic action:

```
move p  =  ask (atomic (do {right 5 ; forward p })) =≪ cows
```

5.4 A HLogo Model Example

A complete model simulating a population of cows living on a field is shown in Fig. 1. In this simple model, Grass grows on random patches in the field and cows move around randomly, eating grass to gain energy. Regrowth of the grass and loss of energy are not included in this simple model. The example code also demonstrates a rudimentary support for visualization: the command `snapshot` can be called at any place in HLogo code to save an image of the current simulation's 2D canvas to a fresh postscript image. The image in Fig. 2 shows a snapshot of a run of the model in Fig. 1. Live visualization, as offered by the NetLogo platform, is part of future work.

6 Parallelizing HLogo

Both HLogo and NetLogo models are compiled to native code to run simulations. HLogo uses the Haskell compiler whereas NetLogo is actually compiled to Java bytecode which is then interpreted by a JVM; however, nowadays JVMs regularly employ Just-In-Time (JIT) compilation to native code. Both HLogo and NetLogo's simulation engines use similar data-structures to store agents: a 2-dimensional array for patches, and tree-based maps for turtles/links. However, they differ fundamentally on how they execute their commanding primitives (ask/of/with). E.g. in NetLogo's ask A c where A is an agentset, the command c is invoked on every agent in A, in a *sequential* manner. NetLogo does provide a variant called ask−concurrent; but this only simulates concurrency by

```
setup = do
    ask (do c ← one_of [green, brown]
            atomic (set_pcolor c)
         ) =≪ patches
    cs ← create_cows 50
    ask (do x ← random_xcor
            y ← random_ycor
            atomic (do set_color white
                       set_energy 50
                       setxy x y
                    )) cs
    reset_ticks

go = forever (do
    t ← ticks
    when (t > 1000) stop
    ask (do {move ; eat_grass}) =≪ cows
    snapshot
    tick)

move = do
    r ← random 50
    atomic (do {right r ; forward 1})

eat_grass = atomic (do
    c ← pcolor
    when (c==green) (do set_pcolor brown
                        e ← energy
                        set_energy (e + 30)))
```

Fig. 1. An example of agent model in HLogo. Cow-turtles move around and eat grass-patches to gain energy.

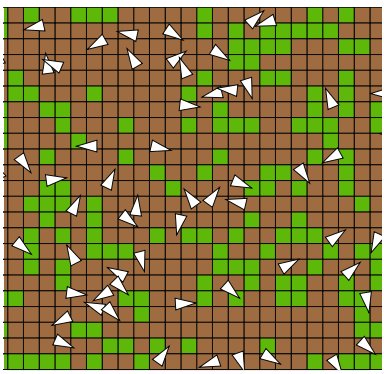


Fig. 2. An example of HLogo visualization output. White triangles represent cows. Green patches are patches with grass. Brown patches have no grass.

interleaving the execution of c between the agents in A. In other words, it does not run the agents in parallel. In contrast, HLogo tries to parallelize the execution of ask/of/with by utilizing Software Transactional Memory (STM) and green threads, two technologies for parallelism provided by Haskell discussed in Sect. 4.3.

Figure 3 shows the algorithm of the worker function behind HLogo's ask. The function $askWorker$ gets the command c to execute and an agentset A. It also gets a few other things as context: $self$ is the id of the agent that calls it and $parent$ is the id of $self$'s parent (in NetLogo also denoted by myself). The worker adopts a 'divide and conquer' strategy: it randomly splits A into N subsets, called $slices$, where N is the number of available CPU cores. For each slice B, a separate green thread t will be created, and the command c will be executed on every agent β in B. So, the execution of the slices is parallel, but within each slice the agents execute sequentially. Executing c on an agent β will need a context similar to the context passed to askWorker, except that β is now the 'self', and the worker's $self$ is its parent. Finally, the worker will wait until all slices finish their execution.

To randomly split A we use a high-quality treefish random generator library [7]. This random generator is splittable, allowing the random generator used by $askWorker$ to be split into N fresh generators, one for each slice so that the threads do not have to compete on a single generator. Note that c may contain another ask, which then needs a random generator to do its own splitting.

Notice that $askWorker$ blocks at the end, to wait until all the threads it spawned have completed. There is also a non-blocking variant of ask named ask_async where the worker simply continues.

$askWorker\ (self, parent)\ c\ A\ =$
  $N \leftarrow$ the number of CPU cores
  $slices \leftarrow$ use $rndG$ to split A into N parts
  $T \leftarrow \emptyset$
  for $B \in slices$
    $t \leftarrow (\lambda() \rightarrow$ **for** $\beta \in B\ \rightarrow$ executeCommand $(\beta, self)\ c)$
    run $t()$ as a new thread
    $T \leftarrow T \cup \{t\}$
    $i \leftarrow i{+}1$
  wait until all threads in T finish

Fig. 3. The algorithm of HLogo 'ask' implementation.

The other two primitives, of_ and with, are implemented in a similar manner, with the only difference being that their workers need to collect (of_) and filter (with) the results of executing c on the agents in A. All three primitives are themselves commands, which implies that their usage can be nested. For example, in:

$$\texttt{ask}\ (\texttt{ask}\ \texttt{eat\_grass} =\!\ll \texttt{cows\_here}) =\!\ll \texttt{green\_patches} \qquad (1)$$

This command will ask every green patch to pass back all the cows that are currently on the patch, and ask these cows to eat the grass there. Note that this nesting will have maximum $N^2 + 1$ green threads running. Haskell runtime system will automatically load-balance the threads to the available CPU cores, if for example there are some patches with less or no cows on them.

6.1 Commands and Transactions

A Haskell thread expects to execute some code with IO effects. Consequently, commands are instances of the `IO` monad. However, if this simple approach is followed, race conditions might occur between the agents. Consider the following HLogo 'procedure' (in Haskell terminology this is a 'function') defining the command `eat_grass`:

```
eat_grass = do g ← grass
               when (g > 30) (do set_grass (g − 30)
                                 e ← energy
                                 set_energy (e + 30))
```

If we allow the commands in `eat_grass`'s body to operate in the IO monad, two race conditions may happen: (a) two cows eat grass from the same patch, but the patch grass level is decreased only once; (b) at another point in the program, an agent 'ask's to (destructively) modify the energy of a currently-eating cow.

Instead, what Hlogo actually does is to store agents' attributes in TVars, i.e. transactional variables as discussed in Sect. 4.3. Basic agent commands (e.g. `right`, `left`, `forward`, but also getters and setters such as `grass` and `set_grass` in the example above) are allowed to execute *only* inside an STM transaction. In other words, these commands are instances of the `STM` monad. As discussed in Sect. 4.2, using the **do**-notation we can compose multiple monadic actions to form a more complicated monadic action. This also applies to STM transactions. This means that the command `eat_grass` is an instance of `STM`. The code in (1) is thus not type correct since `ask` expects an instance of `IO` as the command.

We extend the language with the command `atomic` which given an STM transaction will try to 'run' it; when the `atomic` succeeds, it means its effects have been committed as a whole to the outside (IO) world, and will not be rollbacked. The type of `atomic` is `STM` $a \rightarrow$ `IO` a. So abstractly it is a function that turns an STM transaction into an instance of the `IO` monad. To fix (1) we can do the following, which is now type correct, parallel, and race-condition free (we underline 'atomic' to make it stand out):

```
ask (ask (atomic eat_grass) =≪ cows_here) =≪ green_patches        (2)
```

Does this mean that the programmer should create as large transaction blocks as possible and merely surround them with a single `atomic`? Not exactly, since larger transactions can affect performance negatively, since the probability to conflict with transactions running on other threads increases. If a large transaction has to be rollbacked, the computation it has performed up to the point

of rollback is also wasted. With HLogo, it is left to the programmer to decide if the whole transaction should be atomic or if it is safe to break it into smaller atomic blocks. As an example, below is a variation of `eat_grass` that avoids the above mentioned race-condition (a) and is faster than the variant with top level `atomic` in (2). It does not however avoid the race-condition (b).

```
eat_grass = do g ← atomic grass
               atomic (when (g > 30) (do set_grass (g − 30)
                                         e ← energy
                                         set_energy (e + 30)))
```

Despite the gain in parallelism, STM is not a 'silver bullet' to all problems that occur in a parallel setting. The execution of multiple STM transactions is inherently non-deterministic. In the simplest case, two simultaneous threads competing to modify the same TVar do not always commit in the same order. Consequently, HLogo simulations are non-reproducible, but still consistent with respect to race-conditions. On the bright side, HLogo's engine guarantees that on a 1-core configuration, and if we do not use any asynchronous primitive such as the previously mentioned `ask_async`, the simulation of any agent model is reproducible.

7 Experiment

To compare the performance of HLogo to that of NetLogo, we ran the following benchmarks. They are run on 100×100 patches, forming a torus-shaped canvas. The benchmarks are simulated for 1000 ticks.

1. The benchmark *Redblue* has N turtles. The patches are randomly colored red or blue. At every tick, each turtle moves one step forward, and then turns 30° to the left if it is on a red patch, and else 30° to the right. Agents never write to the same TVar, and therefore their transactions never need to roll back.
2. The benchmark *Cows* has N cows. The patches are seeded with grass. The cows move around randomly and eat grass. Consumed grass will regrow after some random time (but below a certain maximum). Cows compete thus for the grass, so some transactions may conflict and have to roll back.
3. The benchmark *Termites* has N termites. Each patch may contain 1 wood chip. Termites navigate randomly to find a wood chip, pick it up and move it next to other wood chip(s). Later on, sparse areas of wood chips are formed. In this benchmark, termites compete both for picking up and placing of wood chips.
4. The benchmark *Dummy* has N turtles, each simply wiggles randomly and moves. Similar to RedBlue, agents do not conflict, but furthermore they do not interact.

These benchmarks are run on a system provided by the SURF foundation with 32 cores Intel® Xeon E5-2698, 128 GB RAM. Hyper-Threading is disabled since it does not provide true CPU-core parallelism. The OS Ubuntu 16.04 64 bit

was installed, with The Glorious Glasgow Haskell Compiler version 8.0.1 and NetLogo 5.3.1 running Java-8 version OpenJDK 1.8.0_111.

We run them on different configurations, with varying number of CPU cores (1, 2, 4, 8, 16, 32) and the problem size (N). 20 simulations are run for each benchmark and each configuration where we compute the average run time and resident memory. Note that NetLogo first needs to parse and compile the model, and then launches the JVM before it can start running a simulation. Being an embedded DSL, HLogo does not do these. To make the comparison fair, when measuring NetLogo's run time we *exclude* the time it takes to do the aforementioned preparation tasks. The benchmarks results are shown in Table 1: speedup is measured as the ratio of NetLogo execution time over the HLogo execution time for all experiments conducted; additionally, the memory ratio, i.e. HLogo memory used over the NetLogo memory used is given. Figure 4 shows a visualization of the absolute execution time of all four benchmarks.

Table 1. Execution speedup and memory usage of HLogo versions compared to NetLogo for a varying number of processors (cores). N is the size of the problem.

Problem	N	Execution speedup HLogo, #cores						Memory ratio HLogo, #cores					
		1	2	4	8	16	32	1	2	4	8	16	32
RedBlue	1000	0.74	1.18	1.82	2.87	**2.87**	1.33	0.12	0.19	0.3	0.34	0.38	0.44
	2500	0.55	0.67	1.04	1.96	**2.57**	2.33	0.12	0.18	0.25	0.48	0.72	0.72
	5000	0.56	0.65	0.99	1.77	3.01	**3.12**	0.12	0.18	0.27	0.47	0.85	0.98
	10000	0.54	0.67	1.09	2.02	3.24	**4.15**	0.13	0.19	0.28	0.48	0.85	*1.24*
	20000	0.46	0.53	0.82	1.73	3.22	**3.55**	0.16	0.21	0.33	0.48	0.91	*1.69*
	30000	0.40	0.50	1.00	1.82	3.23	**4.71**	0.18	0.26	0.39	0.65	*1.14*	*1.85*
Cows	100	3.29	5.08	7.57	**10.79**	8.99	6.54	0.10	0.17	0.3	0.52	0.82	0.87
	250	2.73	4.00	6.64	**8.57**	8.27	5.97	0.10	0.16	0.29	0.51	0.78	0.88
	500	2.04	3.36	5.19	6.90	**7.09**	6.55	0.10	0.17	0.3	0.54	0.87	*1.03*
	1000	1.55	2.59	4.08	5.37	**6.03**	5.97	0.10	0.16	0.29	0.53	0.88	*1.15*
	2000	1.16	1.84	3.27	4.52	5.28	**5.41**	0.10	0.17	0.29	0.53	0.92	*1.32*
	3000	0.90	1.04	2.28	3.57	4.86	**4.92**	0.10	0.17	0.25	0.46	0.91	*1.77*
Termites	100	1.26	1.72	2.45	**4.29**	3.96	3.03	0.10	0.16	0.26	0.37	0.39	0.43
	250	1.08	1.75	2.69	3.80	5.01	**5.01**	0.09	0.16	0.31	0.51	0.81	0.87
	500	1.09	2.06	3.15	5.17	**7.13**	6.42	0.10	0.17	0.31	0.58	*1.10*	*1.77*
	1000	0.58	1.05	2.04	2.98	4.35	**5.37**	0.10	0.16	0.3	0.57	*1.08*	*2.06*
	2000	0.42	0.84	1.45	2.78	4.61	**6.01**	0.09	0.14	0.27	0.51	0.97	*1.89*
	3000	0.46	0.67	1.17	2.30	4.03	**6.30**	0.09	0.15	0.25	0.48	0.90	*1.5*
Dummy	1000	0.54	0.95	1.24	1.56	**1.75**	1.17	0.25	0.45	0.84	*1.56*	*2.43*	*2.81*
	2500	0.45	0.55	0.90	1.59	**1.95**	1.79	0.22	0.3	0.49	0.99	*1.91*	*3.81*
	5000	0.41	0.51	0.91	1.60	**2.42**	2.38	0.23	0.34	0.52	0.96	*1.86*	*3.64*
	10000	0.40	0.48	0.85	1.58	2.77	**2.96**	0.24	0.32	0.56	0.91	*1.72*	*3.31*
	20000	0.33	0.42	0.70	1.30	2.52	**2.56**	0.28	0.41	0.58	0.99	*1.75*	*3.22*
	30000	0.32	0.53	0.94	1.60	2.39	**3.07**	0.3	0.44	0.68	*1.18*	*2.19*	*4.28*

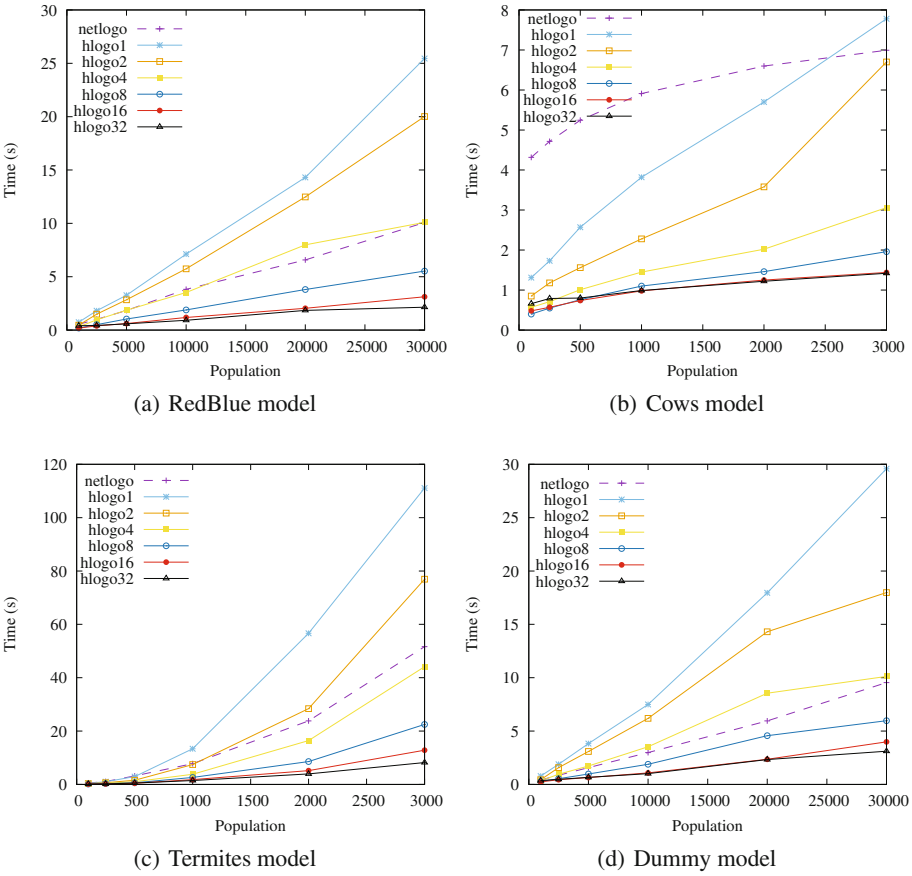


Fig. 4. NetLogo & HLogo execution time for the 4 benchmarks.

Overall, we can clearly witness the speed gain in HLogo as we increase the number of cores, while the performance scalability is retained. HLogo manages to be at its best 10.79 times faster than NetLogo using a configuration of 8 cores. Even with two cores HLogo manages to match or surpass the speed of NetLogo, while using on average 78% less memory, which is a positive outcome considering the fact that STM concurrency incurs certain overhead.

More specifically, the benchmark RedBlue at Fig. 4(a) shows linear growth of the execution time relative to the problem size, which is expected for this model. HLogo manages to be faster than NetLogo on a configuration of 4 cores and above; however, on less cores HLogo's speedup (see Table 1) actually worsens fast, attributed perhaps to the administrative costs of managing and distributing the `turtles` agentset to the different cores.

The results of the Cows benchmark (shown at Fig. 4(b)) indicate a sublinear complexity behaviour, since normally the overall workload grows less than the

number of cows as there is no much grass left to be eaten. Again, the speedup of HLogo is positive for almost all core configurations; however, the speedup degrades (Table 1) as the problem size increases, which is due to increasing conflicts between the cows as they compete for the grass, hence leading to more STM rollbacks.

The Termites benchmark at Fig. 4(c) suggests a superlinear complexity of the model, since the turtles (termites) compete greatly with each other for (dis)placing the wood chips (patches) into forming piles. HLogo manages to be at its best 7.13 times faster than NetLogo using a configuration of 16 cores; the speedup degrades similarly to the Cows benchmark, attributed to the vast competition between the turtles.

The Dummy benchmark of Fig. 4(d) shows analogous complexity behaviour (linear) to the RedBlue benchmark. The HLogo speedup benefit is mostly positive but less than what is offered by RedBlue, although the problem has less agent interaction. This can be attributed to the fact Dummy's model has one large atomic block, whereas RedBlue has finer-grained atomic blocks. For the latter, this leads to smaller-sized STM logs and less traversal of the logs' contents when committing each STM log, i.e. atomically writing out the effects to memory.

HLogo, the above benchmarks, and other examples are available as open-source software at http://github.com/bezirg/hlogo.

8 Conclusions and Future Work

We have presented HLogo, a variant of the ABMS framework NetLogo, that offers an embedded DSL front end. At the back end it utilizes Software Transactional Memory (STM) to obtain performance gains from multi-core execution, and at the same time hide the complexity of concurrent programming from the programmers. As an embedded DSL, HLogo has the advantage of inheriting the Haskell's module system, thus allowing its programmers to import any of the whole wealth of Haskell libraries. Furthermore, the DSL is statically strongly typed for all its expressions and agent commands, which adds a certain level of safety when crafting a model.

Our benchmarks showed that HLogo's performance does not suffer from the use of STM. In fact, on multi CPU cores HLogo runs faster than NetLogo. Giving HLogo more cores progressively increases its speed. HLogo also uses less memory than NetLogo, up to a certain number of CPU cores (8 in most of our experiments' configurations). On the other hand, the trade off is that HLogo simulations on multiple cores are not reproducible. Despite this, we believe that there is room for applying HLogo, namely on problems where reproducibility is not a factor, and where speedup is crucial to keep the running time feasible.

As a final note, we want to add that HLogo's STM-based parallelism is in principle framework-agnostic and thus could be applied to other ABMS frameworks. For example, it can be applied to NetLogo. It should be possible to extend NetLogo with the `atomic` construct. Then, one can use one of available STM implementations in Scala (the implementation language of NetLogo) and

mimic the described Haskell implementation. This might be attractive to the already large NetLogo community, thus giving NetLogo the performance benefits of HLogo.

Future Work. Our work so far has mainly focused on providing a front end ABMS DSL and its backend simulation engine. To improve HLogo's usability it will need a decent visualization front end, e.g. as NetLogo now has and this is a future work direction. Another feature that HLogo currently lacks and would come as a great addition is parameter sweeping, i.e. executing the model with multiple runs, each with a different parameter input (similar to NetLogo's BehaviorSpace tool).

With respect to its simulation engine, HLogo currently splits the workload to threads using a random divide and conquer strategy. If transactions that write to some common TVars are distributed to different CPU cores, this may lead to transactions conflicts and therefore rollbacks. On the other hand, assigning them to the same CPU core will avoid rollbacks. A smarter dynamic workload distribution strategy should take this into account. Such a strategy could be based on for example the turtles' last known positions, or how they are connected by links. Turtles that are close to each other, e.g. linked together, are more likely to conflict. A static approach is probably less likely to be successful because turtles move around, and links can be added and removed dynamically.

On the technical side, the turtles and links agentsets can be modified, currently, only in a non-concurrent fashion: a thread has to acquire a lock on the agentset to create or remove (`die`) a turtle or link. By changing the data structure to a concurrently modifiable tree-map implementation we would benefit from faster in-parallel insertions and deletions of turtles and links. Furthermore, we can consider optimizing the tree-map data structure used to store the turtles (see the beginning of Sect. 6) with a hybrid representation where turtles are added in-order into a `who`-indexed vector, which *if needed* is transformed into a sparse tree-map implementation when removals happen (as in 'die') dynamically at runtime. For the patches, we can consider low-level optimization of their array data structure by storing the indices (patches) in the so-called Z-order scan (similar to a zigzag) instead of linearly as it is now. A Z-order patch array would result in better performance because of better data locality: agents most often interact with 2D-neighbouring patches and would then store the data of the neighbouring patches close to each other, leading *on average* to less cache misses than accessing the array in the common row order.

HLogo currently requires programmers to specify the atomicity level of the agents by inserting calls to the function atomic. Through this mechanism, the programmers can increase parallelism e.g. by grouping an agent's access to unrelated TVars to different atomic blocks. As future work, we want to investigate if the function atomic can be extended e.g. to log information that would enable a simulation run to be reproduced or at least to some degree reconstructed, even if it was originally run on multiple cores. The challenge here is to be able to log enough information without slowing down the simulations. We also want to

investigate if the insertions of atomic can be done automatically, e.g. through data flow analysis. Moreover, some STM transactions can be accelerated after applying certain optimizations, e.g. a wiggling cow move:

```
atomic( do{right =<< random 50 ; left =<< random 50})
```

can be optimized to `atomic( do{i ← random 50 ; j ← random 50 ; left(i+j)})` which is faster since the STM transaction log is shortened through combining two modifications of the turtle's heading (right, left) to one (left). The program remains consistent since this code runs atomically: no other agent could have, in anyway, witnessed the intermediate modification.

Finally, since Cloud computing has become widely available, it might also be interesting to investigate if HLogo can be extended to a distributed setting. For example, this would enable HLogo models to run in High-performance Computing (HPC). There is also the extreme case where models cannot fit in a single shared memory machine and have to be distributed to multiple processing nodes. There are Haskell technologies, such as Distributed Software Transactional Memory [20] or Cloud Haskell [12] that can be employed towards this direction.

Acknowledgements. This work was partially funded by the EU project FP7-610582 ENVISAGE: Engineering Virtualized Services (http://envisage-project.eu). All the benchmarks in this work were carried out on the Dutch national HPC e-infrastructure, kindly provided by the SURF Foundation (http://surf.nl).

References

1. Haskell, an advanced, purely functional programming language. https://www.haskell.org/
2. Grimm, V., et al.: A standard protocol for describing individual-based and agent-based models. Ecol. Modell. **198**(1–2), 115–126 (2006)
3. Berndsen, J.: The Hasim package. https://hackage.haskell.org/package/hasim
4. Bezirgiannis, N., Prasetya, I.S.W.B., Sakellariou, I.: HLogo: a parallel Haskell variant of NetLogo. In: Proceedings of the 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, SIMULTECH, pp. 119–128. SciTePress (2016)
5. Bjesse, P., Claessen, K., Sheeran, M., Singh, S.: Lava: hardware design in Haskell. In: Proceedings of the 3rd ACM SIGPLAN International Conference on Functional Programming. ACM (1998)
6. Castle, C.J.E., Crooks, A.T.: Principles and concepts of agent-based modelling for developing geospatial simulations, September 2006
7. Claessen, K., Pałka, M.H.: Splittable pseudorandom number generators using cryptographic hashing. In: ACM SIGPLAN Notices, vol. 48, pp. 47–58. ACM (2013)
8. Deissenberg, C., van der Hoog, S., Dawid, H.: EURACE: a massively parallel agent-based model of the European economy. Appl. Math. Comput. **204**(2), 541–552 (2008)
9. Discolo, A., Harris, T., Marlow, S., Peyton, Singh, S.: Lock-free data structures using STMs in Haskell, April 2006

10. D'Souza, R.M., Lysenko, M., Rahmani, K.: SugarScape on steroids: simulating over a million agents at interactive rates (2007)
11. Elliott, C.: Functional images. In: The Fun of Programming. Cornerstones of Computing. Palgrave, Basingstoke (2003)
12. Epstein, J., Black, A.P., Peyton-Jones, S.: Towards Haskell in the cloud. In: ACM SIGPLAN Notices, vol. 46, pp. 118–129. ACM (2011)
13. Epstein, J.M., Axtell, R.: Growing Artificial Societies: Social Science from the Bottom Up. Brookings Institution Press, Washington, D.C. (1996)
14. Gill, A.: Domain-specific languages and code synthesis using haskell. Queue **12**(4), 30 (2014)
15. Grimm, V., Revilla, E., Berger, U., Jeltsch, F., Mooij, W.M., Railsback, S.F., Thulke, H.H., Weiner, J., Wiegand, T., DeAngelis, D.L.: Pattern-oriented modeling of agent-based complex systems: lessons from ecology. Science **310**(5750), 987–991 (2005)
16. Hybinette, M., Kraemer, E., Xiong, Y., Matthews, G., Ahmed, J.: SASSY: a design for a scalable agent-based simulation system using a distributed discrete event infrastructure, pp. 926–933 (2006)
17. Kiran, M., Richmond, P., Holcombe, M., Chin, L. S., Worth, D., Greenough, C.: FLAME: simulating large populations of agents on parallel hardware architectures. In: International Foundation for Autonomous Agents and Multiagent Systems, pp. 1633–1636 (2010)
18. Knight, T.: An architecture for mostly functional languages. In: LFP 1986, pp. 105–112. ACM, New York (1986)
19. Koehler, M., Tivnan, B., Upton, S.: Clustered computing with Netlogo and RepastJ: beyond chewing gum and duct tape (2005)
20. Kupke, F.K.: Robust distributed software transactions for Haskell. Ph.D. thesis, Christian-Albrechts Universität Kiel (2010)
21. Logan, B., Theodoropoulos, G.: The distributed simulation of multiagent systems. Proc. IEEE **89**(2), 174–185 (2001)
22. Massaioli, F., Castiglione, F., Bernaschi, M.: OpenMP parallelization of agent-based models. Parallel Comput. **31**(10), 1066–1081 (2005)
23. Milner, R.: A theory of type polymorphism in programming. J. Comput. Syst. Sci. **17**(3), 348–375 (1978)
24. North, M.J., Collier, N.T., Ozik, J., Tatara, E.R., Macal, C.M., Bragen, M., Sydelko, P.: Complex adaptive systems modeling with repast simphony. Complex Adapt. Syst. Model. **1**(1), 1–26 (2013)
25. Perfumo, C., Sönmez, N., Stipic, S., Unsal, O., Cristal, A., Harris, T., Valero, M.: The limits of software transactional memory (STM): dissecting Haskell STM applications on a many-core environment. In: CF 2008, pp. 67–78. ACM (2008)
26. Peterson, J., Hager, G.: Monadic robotics. In: Proceedings of the 2nd Conference on Domain-Specific Languages, DSL 1999, pp. 95–108. ACM, New York (1999)
27. Peyton Jones, S.L., Wadler, P.: Imperative functional programming. In: Proceedings of the 20th Symposium on Principles of Programming Languages, POPL. ACM (1993)
28. Pogson, M., Smallwood, R., Qwarnstrom, E., Holcombe, M.: Formal agent-based modelling of intracellular chemical interactions. Biosystems **85**(1), 37–45 (2006)
29. Railsback, S.F., Lytinen, S.L., Jackson, S.K.: Agent-based simulation platforms: review and development recommendations. SIMULATION **82**(9), 609–623 (2006)
30. Riley, P.F., Riley, G.F.: Next generation modeling III - agents: spades—a distributed agent simulation environment with software-in-the-loop execution. In: WSC 2003, Winter Simulation Conference, pp. 817–825 (2003)

31. Sakellariou, I., Kefalas, P., Stamatopoulou, I.: Enhancing NetLogo to simulate BDI communicating agents. In: Artificial Intelligence: Theories, Models and Applications. LNCS, vol. 5138, pp. 263–275. Springer (2008)
32. Salamon, T.: Design of Agent-Based Models. Eva & Tomas Bruckner Publishing, Repin (2011)
33. Shavit, N., Touitou, D.: Software transactional memory. In: PODC 1995, pp. 204–213. ACM (1995)
34. Sheard, T., Jones, S.P.: Template meta-programming for Haskell. SIGPLAN Not. **37**(12), 60–75 (2002)
35. Sorokin, D.: Aivika. http://www.aivikasoft.com/en/products/aivika.html
36. Tobias, R., Hofmann, C.: Evaluation of free java-libraries for social-scientific agent based simulation. J. Artif. Soc. Soc. Simul. **7**(1), 6 (2004)
37. Van Deursen, A., Klint, P., Visser, J.: Domain-specific languages: an annotated bibliography. Sigplan Not. **35**(6), 26–36 (2000)
38. Wilensky, U.: NetLogo (1999)
39. Wilensky, U.: Statistical mechanics for secondary school: the GasLab multi-agent modeling toolkit. Int. J. Comput. Math. Learn. **8**(1), 1–41 (2003)
40. Wilkerson-Jerde, M., Wilensky, U.: Restructuring change, interpreting changes: the deltatick modeling and analysis toolkit (2010)

Requirements Gathering and Validation for Risk-Oriented Tool Support in Supply Chains

Stephan Printz[1], Christophe Ponsard[3(✉)], Johann Philipp von Cube[2], Renaud De Landtsheer[3], Gustavo Ospina[3], Philippe Massonet[3], Robert Schmitt[2,4], and Sabina Jeschke[1]

[1] Institute for Management Cybernetics (IfU), RWTH Aachen University, Aachen, Germany
{stephan.printz,sabina.jeschke}@ifu.rwth-aachen.de
[2] Fraunhofer Institute for Production Technology (IPT), Aachen, Germany
{philipp.von.cube,robert.schmitt}@ipt.fraunhofer.de
[3] CETIC Research Centre, Charleroi, Belgium
{christophe.ponsard,renaud.delandtsheer,gustavo.ospina,philippe.massonet}@cetic.be
[4] Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University, Aachen, Germany

Abstract. Managing risks in supply chains is challenging for most companies, given that the globalisation process is strengthening production constraints and also introducing more procurements risks. This is even more difficult for smaller companies because of their lack of resources to develop specific expertise or buy expensive tools. In order to be successful, a project aiming at improving the state of practice in this area must address two key activities: gaining a good knowledge of the actual needs and validating the results. This paper reports about the process followed for supporting those activities using an agile approach. It relies on an initial survey conducted in companies, mostly from the manufacturing domain in Belgium and Germany together with the deeper involvement of 10 companies which provided concrete requirements directly linked with validation cases. We present the main outcome of the requirements gathering process, especially the survey analysis, as well as the lessons learned about our iterative validation process.

Keywords: Discrete Event Simulation · Manufacturing · Supply chain · Procurement risks · Risk management · Validation

1 Introduction

Supply chain risk management (SCRM) is the implementation of strategies in order to manage both everyday and exceptional risks throughout the supply chain. This can be achieved through a continuous risk assessment aiming at reducing the number of vulnerabilities, thus ensuring continuity [1]. Such risks

M.S. Obaidat et al. (eds.), *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing 676,
https://doi.org/10.1007/978-3-319-69832-8_8

can occur for several reasons, both of external nature (procurement risks of geographic, political, social nature, etc.) and internal to the company (machine reliability, nature of specific operations, etc.). The Risk management (RM) is performed by either qualitative or quantitative models [2].

Supporting companies to take the right decisions in the face of risks is a challenging task. Small and medium enterprises (SMEs) are particularly challenged because of limited resources they can devote to this task, despite the fact that failing to address risks could dramatically affect their business. The ultimate goal of our research is to produce a user-friendly and tool-supported methodology that will guide the user through the whole risk assessment process, as shown in Fig. 1.

In order to support our research, it was important to fully characterise current practices of SMEs with respect to supply chain risks by getting answers to following questions:

- What are the risks perceived by companies?
- How do they rank such risks in terms of importance, taking into account both likelihood and impact?
- How do they manage such risks in terms of people and tools?
- What do they require of methods and tools in order to integrate them?

In this paper, we report on the process followed to answer those questions, based on two complementary kinds of activities:

- an initial survey conducted in the manufacturing sector across Belgium and Germany to understand the state of practice and the main needs.
- iterative validations conducted during the project with a user committee, giving feedback on successive refinements of the tool prototype and enabling a deeper understanding of SMEs needs.

In addition to findings specific to risk management for supply chains, this paper also presents some lessons learned from the process. More specifically, it stresses the importance of a number of non-functional requirements such as usability, learning ability and security for a successful adoption by SMEs.

The paper is structured as follows: Sect. 2 gives some background related to risk and risk management, Sect. 3 presents the survey process including an overview of the participating companies, procurement aspects, risk perception and tool related requirements. Section 4 describes the validation process, the resulting refined requirements as well as the lessons learned during this process. Finally, in Sect. 5 a conclusion is drawn and some recommendations for a similar tool development process are given.

2 Background on Risk

2.1 Notion of Risk

Supply Chain Risk impacts every organization irrespective of sector, size or location in the supply chain [3]. The notion of risk is, of course, more general and all

companies have to address risks that can affect their business, i.e. they need to develop adequate strategies to face events likely to occur with undesired consequence. Defining risk involves thus two key components: the probability that an undesired event occurs and its consequence's magnitude. Those aspects are part of all risks definitions, e.g. the ISO 31000 standard about risk management defines risk as the impact on uncertainty to objectives [4]. Nevertheless, the first quantitative risk assessment approach was defined by Bernouilli in 1738 [5] as the mean value of the undesired consequences (L_i) weighted by their likelihood $p(L_i)$. So the total risk can be written as: $R_{total} = \sum L_i * p(L_i)$

Undesirable events can be identified using a variety of techniques which are extensively described in referenced books [6,7]. This includes informal techniques such as brainstorming and check-lists to more structured and systematic techniques like FTA (Fault-Tree Analysis), FMEA (Failure Mode Effect Analysis), HAZOP (HAZard OPerability), RCA (Root Cause Analysis). Although, the later mentioned techniques are likely to be more effective, a fundamental issue is that risk identification is never complete [4].

Risk likelihood can be modelled with probability distributions [8], as the occurrence of a risk hazard in a process or system is *uncertain*. In [9], a theory of probabilistic risk analysis is developed, which is associated with the concept of system reliability. As a risk is defined as deviation from a target value, statistical measures can be applied to operationalise and compare possible magnitudes of such deviations [10]. Evaluation of the risk analysis and the reliability of a system can be done with the Monte-Carlo method [11].

Assessing the likelihood of such events is challenging task. The assessment of the likelihood may rely on past observations either common to a domain (e.g. mean time before failure time of some machine) or directly collected as part of the company operation (e.g. reliability of a supplier to deliver in time) and moreover on simulation techniques. For complex open systems, a system dynamics approach is recommended [12].

2.2 Risk Management

The basic process of risk management (Fig. 1) is described in standards ISO 31000 and ONR 49000 ff. IEC 31010 provides an overview of corresponding risk management methods and techniques for a specific process.

The main objective of risk management lies in the assessment of major corporate goals in regards to risk policy strategies. Hence, risks affecting long lasting business success need to be controlled. However, enterprises will never be able to totally eliminate risks and will always have to consider a certain degree of residual risk [13]. One key task of risk management is to identify and analyse risks as early as possible in order to take cost optimal risk treating actions [3].

The objectives to be assessed are, for instance, strategic, organisational, and related to projects, products or processes. However, Heckmann pointed out that there is no common definition of SCRM [14]. This literature review provides an overview of several classes of risks.

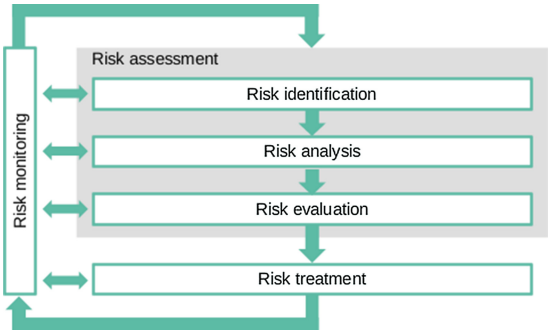


Fig. 1. Risk management process according [4].

In order to reduce complexity, an aggregation of three risk classes was performed, which considers transportation and warehouse risks in the context of manufacturing.

- *Quantity risks* are related to how lack or excess of materials (from raw materials to produces) affect a manufacturing process.
- *Quality risks* are related to the good or bad conditioning of materials as well as the respect of specifications for internal quality. Finally,
- *Delay risks* concern the time aspects, especially for the supply of materials, the processing time and the transport from/to warehouses.

Those classes are detailed in Table 1 and are also widely reported in the literature [15–22].

Table 1. Risk classification.

Risk class	Definition	Root cause
Quantity	Risks leading to deviations in the disposed quantity	Insolvency, storage, order cycle, sourcing strategy, supplier, order strategy
Quality	Risks regarding the quality of supplied goods	Processing, sourcing strategy, supplier, logistics
Delay	Risks causing unscheduled deviations	Processing, logistics, delivery time, transportation capacity, number of brokers/transfer points

The quantitative assessment of supply chain risks is evaluated using the probability of an undesired event and its expected consequence. For instance, Ziegenbein extended the approach to the number of suppliers and interruption time. However, this approach is a mathematical model. There is no connection to the process and value added chain [23].

3 Survey on Risk Management in Supply Chain SMEs

3.1 Survey Process

The survey was carried out between October 2014 and mid-2015. It was based on a trilingual form (French, German, English) that was distributed to companies in Belgium (mostly in Wallonia) and Germany through different communication channels, such as dedicated mailing lists and social networks. The geographical factors were determined by the collaborative SimQRi project, which involved a representative set of industrial SMEs and their focus on risk assessment [24].

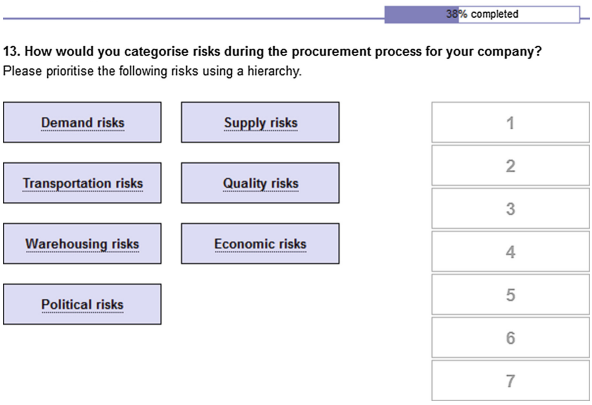


Fig. 2. Example of question [25].

The survey was composed of about 40 questions in total and had different sections: one to understand the company size and business, one to understand the importance of the procurement process, another one to identify the current way in which risks are managed, and finally one to determine the requirements necessary for better tool support. Figure 2 illustrates a typical question, designed to be simple to understand and answer. The indicative time needed to answer the survey is about 15 min. The survey was available through a dedicated website.

3.2 Characterisation of Participating Companies

Around 70 companies answered the invitation and despite their answers being anonymous, we were able to record the contact data of the companies interested in following the project and wanting to get more involved in the process through a user committee. The initial user committee was also the first target group used to fine-tune the survey before it was released to a wider audience. The average age of the participants was 45 years, 9 were female and 61 male, with an average number of 15 years of professional experience in risk management.

A global overview of the whole sample is depicted in Fig. 3. The number of participating companies was balanced between Belgium and Germany (given the size of the activity sectors in both countries). A great variety of manufacturing industrial sectors were covered, with no predominance of any specific sector. The majority of the companies were medium-sized (between 50 and 250 employees), though Walloon companies tended to be smaller, which corresponds well to their economic make-up. About one third (26 participants) had a position associated with risk management, 21 participants are not compelled to carry out risk management but do so anyway, 13 participants have a little experience with risk management but are interested, and 5 participants have no experience with risk management at all.

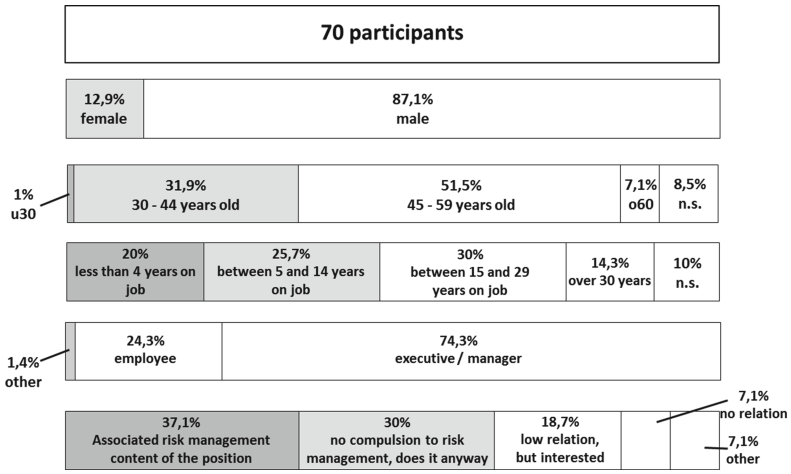


Fig. 3. Main characteristics of participating companies [25].

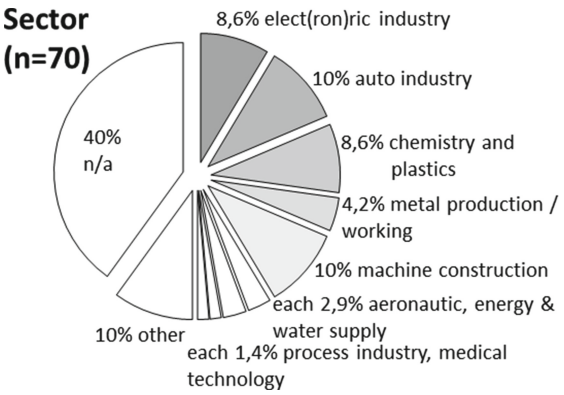


Fig. 4. Sectors represented in the survey [25].

The main sectors of activities are shown in Fig. 4. The automotive industry (10%) and machine construction (10%) are leading, followed by electric/electronic industry (8,6%), chemistry/plastics (8,6%) and the metal production/working (4,2%). Other sectors are less represented. Over 58 of the companies participating are located in Germany, five in Belgium, one in the Netherlands, and three are from other countries in the European Union (EU). Three other companies are located outside of the EU. Globally this is consistent with the survey area and relative importance of the sectors within that area.

The size and turnover results present quite a similar profile, as shown in Fig. 5. Less than four companies have fewer than 50 employees ("small" size), 25 companies have up to 250 employees ("medium" size) and 13 companies are over 250 employees.

Finally, regarding to procurement risks more specifically, for the most part the participating companies were manufacturers of final products (60% of answers), however there was also a significant number of part assembly

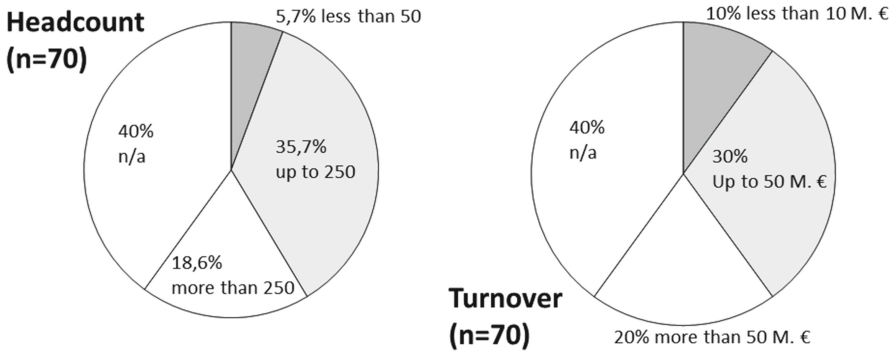


Fig. 5. Size and turnover of the participating companies [25].

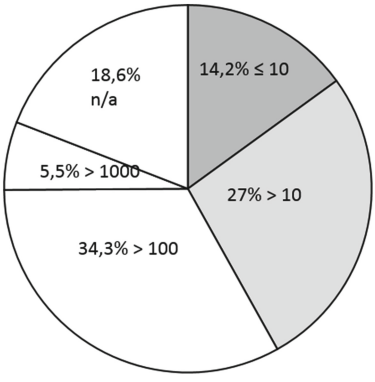


Fig. 6. Distribution of the number of suppliers [25].

companies (25%), as well as part suppliers (15%), though to a lesser extent. With respect to the number of suppliers for each company, Fig. 6 shows the average of suppliers is quite high. Interestingly, there were as many companies present with fewer than 100 suppliers, as companies with more than 100 suppliers. This calls for methods and tools able to manage an important supplier base.

3.3 Risk Management

Asked where risk management takes place in the company, 31 participants named the "executive board", 18 participants "supply chain management", and 5 participants "logistics". Whilst 5 participants chose "other sections", 10 did not specify at all. Half of the participants do not prioritise risks, 57% do not even have a system for the categorisation of risks, all of which is depicted Fig. 7.

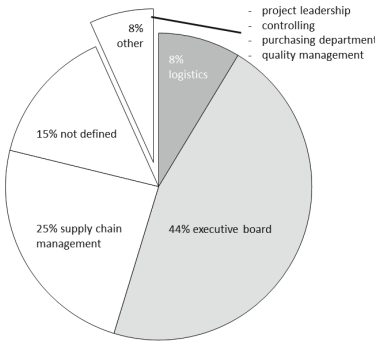


Fig. 7. Function in charge of risk management [25].

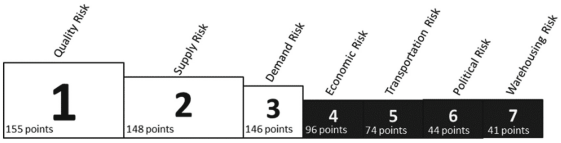


Fig. 8. Risk prioritisation in global process [25].

To set up a priority hierarchy of risks relevant to global manufacturing processes, we asked the companies to rank their top 3 risks. Figure 8 shows quality risks (products that cannot fulfil quality requirements), supply risks (constraints on the volume and the delays required by the clients), and risks related demands (which are directly related to procurement).

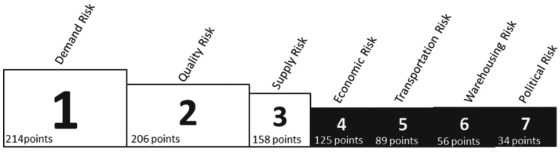


Fig. 9. Risk prioritisation for procurement process [25].

Considering procurement processes (before manufacturing) more specifically, Fig. 9 shows a similar top 3 risks, with procurement risks logically ranking first. Other risks that intervene, though to a lesser extent, are economical risks (e.g. bankruptcy of a supplier), political risks (related to the political situation of a country or region), transport risks (possibility of losses or delays in conveyance) and storage risks (losses or stocking degradation).

3.4 Need for Better Risk Management Tools

The survey reveals that the majority of SMEs does not have any kind of risk management tool or, more precisely, that they rely on standard office tools, like spreadsheets. Barely 10% of companies have dedicated tools for risk management. In this section, we present a summary of requirements identified by direct questions about better tool support.

Regarding the Risks that Require More Support. Figure 10 shows the same top 3 as those identified in the previous section, which is quite consistent with the importance of those risks. Over 50% chose quality risk and demand risk, while over 80% do not see the benefit in receiving support in assessing political and warehousing risks.

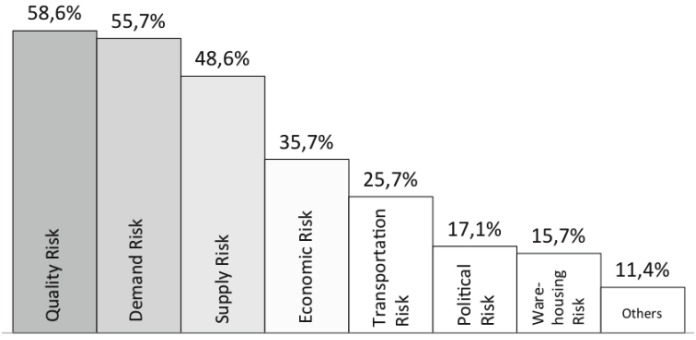


Fig. 10. Tool support by risk categories [25].

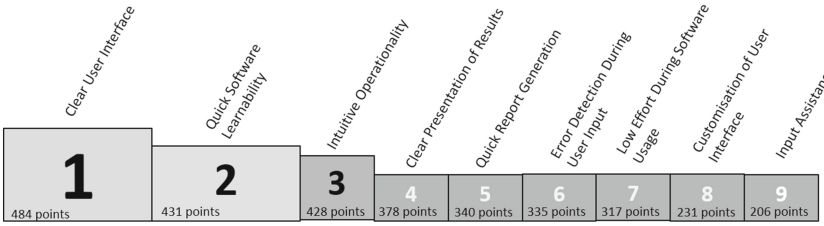


Fig. 11. Non-functional requirements for tool support [25].

Regarding the Most Important Tool Requirements. Figure 11 shows how key requirements for SME oriented tooling were ranked. Three clearly emerging requirements at about the clarity of the user interface, the learning curve and the intuitive use. Those are actually quite interrelated.

3.5 Fine-Grained Requirements Identification Using Correlation Analysis

The support of assessing and simulating external risks is a challenging task. In order to identify interrelations between surrounding risks and desired support of risk management, a correlation analysis was performed. A computation of the Pearson correlation was carried out, in order to check the significance (s) of the SMEs status and expectations on the current risk management tools and simulation. The following tables show an extract of the computed Pearson correlation. Beside the variable itself, the Number of answers (N) as well as the correlation coefficient (cc) and significance (s) are given. The significance is expressed by two different levels:

- *: The correlation is significant at the level of 0.01 (2-sided)
- **: The correlation is significant at the level of 0.05 (2-sided)

In Table 2 the impact of the prioritised risk classes beside the position in the supply chain, the information sharing and data protection are represented. The correlation identifies an increasing impact of risks, namely quality ($cc = 0.439^{**}$, $s = 0.001$), supply ($cc = 0.338^{**}$, $s = 0.009$) and transport risk on the company. This means, the impact of risks would decrease, if you get support by risk management tools and especially simulation. Instead, the correlation reveals the high dependency of external effects on the vulnerability of a supply chain and the need of accounting external risks in the software. However, these correlations match with the findings of desired support for relevant risks by a risk management tool. With respect of the other top ranked quality and supply risk (cf. Fig. 10) **the correlation analysis confirms the importance of the supporting manager by assessing those risks with an adequate risk management tool**.

Another finding is the negative correlation regarding the position in the supply chain ($cc = -0.429^{**}$, $s = 0.007$). Usually, with increasing tier in the supply

chain, the higher the vulnerability. The reason is the increased complexity by the use of more downstream suppliers. **In fact, there is a low need for support by risk management tool with increased tier in the supply chain**. Furthermore, the correlation analysis of the sharing information with other companies and data protection shows significant influence in risk management. For this purpose, **risk management tools should integrate the functionalities of information sharing and data protection**.

Table 2. Correlation with scalability: support of assessing and simulating risks.

Variable	N	cc	s
Impact quality risk	58	0.439**	0.001
Impact supplier risk	58	0.338**	0.009
Impact transport risk	58	0.310*	0.018
Position in supply chain	38	−0.429**	0.007
Ease the sharing of procurement risk analysis and simulation across company members	58	0.432**	0.001
Protection of company data about risk information and procurement process	58	0.311*	0.018

A major issue in risk management are the interrelation between risks and the difficulty of their assessment, also no correlation for top 3 risk categories of quality, demand and supply risks could be derived (cf. Fig. 9). However, the highest correlation and significance affects warehouse risks (cf. Table 3). **In contrast to quality risks, warehouse risks are a not favoured risk category regarding risk management tool support**. However, there is an impact on warehouse risks and the simulation of surrounding risks in general. A significant correlation of quality (cc = 0.237*, s = 0.048) and supply risks (cc = 0.223, s = 0.063) emerge.

Table 3. Correlation with risk management.

Variable	N	cc	s
Difficulty warehouse risk	70	0.239*	0.046
Difficulty quality risk	70	0.237*	0.048
Difficulty supply risk	70	0.223	0.063

After a more general view on risk management and simulation, an overview of the software and their interrelations to several parameters (non-functional, functional) is given. If participants are using a risk management software, they had to rate the software by several aspects. That is why only few people answers those questions referenced in Table 4. Nevertheless, users of current risk management software, rate the overall existing software less good (negative correlation coefficient), when the number of suppliers of a company increases (N = 7,

cc = −0.835*, s = 0.019). **The operation of the risk software seems thus to be perceived more complex when managing a high number of suppliers**. This might result of different causes like intrinsic complexity of such scenarios or an unclear presentation of the graphical user interface which should be addressed in the software analysis and development.

Table 4. Correlation with overall assessment of the software.

Variable	N	cc	s
Number of suppliers	7	−0.835*	0.019

In contrast, the correlation between the rating of the clearness of the user interface and the number of suppliers show positive correlation and significance (cf. Table 5). This is due to the fact, that **there is a more important need for a risk management tool with increasing number of suppliers and this is not linked to the performance at all**. With increasing number of suppliers there is a need for clearness of the software's user interface (N = 7, cc = 0.770*, s = 0.043). Not surprisingly, the software is easy to learn, when the clearness of software's user interface increases. Thus, it is mandatory to account the layout of the user interface during software development.

Table 5. Correlation with clearness of software's user interface.

Variable	N	cc	s
Number of suppliers	7	0.770*	0.043
Software is easy to learn	8	0.745	0.034

After presenting "non-functional" requirements and their correlation as well as significance, the "functional" requirements are examined in Table 6. **Current users of a risk management tool state that quality risk assessment with adequate time effort is difficult** (N = 8, cc = 0.721*, s = 0.044). Anyhow, due to a lower number of answers, there is a need of more reliable data for statistical evidence. In the performed computation only such statistical evidence showed up for this prioritised risk class (cf. Fig. 9) which is a desired support risk category, too (cf. Fig. 10).

Table 6. Correlation with comparatively short time risk assessment possible.

Variable	N	cc	s
Difficulty quality risk	8	0.721*	0.044

Moreover, Table 7 shows evidence, that the **developed software should be usable without expertise or training programs. A reasonable value of performance, regarding minimal economic effort and less time consumption is desired**. If software recognises input errors and reports them to the user, the user does not need the help of experts for interpreting simulation results analysis (N = 8, cc = 0.771*, s = 0.025).

Table 7. Correlation with specific advice to avoid input errors by the software.

Variable	N	cc	s
The softwares analysis results are reasonable without expertise	8	0.771*	0.025

Table 8. Correlation with integration: automate data export from company procurement and supply chain systems.

Variable	N	cc	s
Number of employees	38	0.355*	0.029
Turnover	38	0.451**	0.004
Ease the sharing of procurement risk analysis and simulation across company members	58	0.482**	0.000
Protection of company data about risk information and procurement process	58	0.266*	0.044
Support heavy simulation on external infrastructure	58	0.518**	0.000

Finally, about the need for integration, the results of correlation analysis shown in Table 8 confirm that **an automated data export from companies procurement and supply chain systems (ERP-System) is mandatory and that the bigger the company, the more important this requirement**. Moreover, the sharing of simulation across company members is related to less effort (0.482**, s = 0.000). But with automated data export the protection of the company data have to be considered (cc = 0.266*, s = 0.044), as well as other fields of application/opportunities, such as simulation of external effects on the company itself (cc = 0.518**, s = 0.000).

4 Agile Requirements Validation Process

The results expressed in the survey have to be taken with care, because questions can be misinterpreted and their statistical significance is not always very high especially when filtering on specific sub-samples. As the questions were mostly closed there can be some bias and important requirements can also be missed. For this reason, it was important to ensure the elaboration of requirements more in

deep. An efficient technique to achieve this in a software development project is to rely in an Agile approaches, where successive versions of the tool are produced on a regular basis in order to get some form of feedback from the end-users [26]. In a first draft, the tool has a very partial demonstration, also later the users might start to use the tool themselves. At each step requirements are revisited and new requirements might be discovered. In this section, we describe the approach followed and a summary of the resulting tool requirements.

4.1 Overview of the Agile Process Followed

The project execution was organised at two levels:

- five major iterations of six months duration (except the last one) aimed at releasing an artefact that could undergo so form of validation with end-users and, based on the collected feedback, to more precisely define the goals of the next iteration.
- each iteration was divided in short technical sprints of typically two to maximum four weeks with internal demonstration and debrief usually through teleconference.

Table 9 shows a summary of the main goals, artefacts and validation feedback that was produced during the five major iterations of the project execution.

Table 9. Major iterations of the Agile process followed.

Iteration	Goals	Artefact produced	Validation feedback
Semester 1	Domain survey Technical foundations	Requirements Simulation framework	Requirements amended by user committee
Semester 2	Tool architecture	First prototypes for editor and simulator	User interface requirements from experts Need to enrich modelling and query language
Semester 3	Integration and web deployment	Web version on-line	Need for process support Usability suggestions Security concerns Integration needs
Semester 4	Addressing needs for process support, security and integration	Web UI with risk wizard Desktop-based UI	More usability suggestions Need for report generation Missing user manual
Extension (3 months)	Report generation Bundling	Final version of web and desktop tools	Further needs about report template, ERP integration, risk parameter estimation

4.2 Summary of Functional and Non-functional Tool Requirements

This section presents a summary of the consolidated list of requirements at the end of the project, including initial requirements gathered during the survey and various updates collected during the project iterations with the user committee. The requirements were sorted in the two major categories of functional and non-functional requirements [27]. For each category a table is produced to detail the main requirement types, some representative requirements and the time the requirement was identified. Those tables are discussed in the next section.

Functional Requirements. They are detailed in Table 10. The users wishes for a tool providing support in the risk analysis process, starting with risk identification and the elaboration of a risk-oriented model that can be simulated using the Monte-Carlo simulation, too. During the simulation, specific probes are used to compute risk related queries into the model in a statistical way. The simulation results can be analysed in direct relation to the risks, all of which is presented on a dashboard. The effects of specific measures can then be considered and simulated again in order to control the significant risks.

Table 10. Functional requirements.

Req. type	Requirements	Introduced
Process modelling	Process API	Semester 1
	Graphical web editor	Semester 1
Risk identification	Probability distributions	Semester 1
	KPI definition, query language	Semester 2
	Risk wizard	Semester 3
Risk simulation	Discrete Event simulation	Semester 1
	Monte Carlo simulation	Semester 1
	System dynamics simulation	Semester 2
	Process attributes support	Semester 3
Risk analysis and mitigation	Trace analysis tool	Semester 2
	Risk dashboard, ABC Analysis	Semester 3
	Sweep on parameter	Semester 3
	Detailed reports	Semester 4

Non-functional Requirements. They are summarised in Table 11. Such requirements are very important for user adoption. A key requirement is that the manufacturing processes should be captured through a web-graphical editor. This interface should be designed with usability and ease of installation in mind and it will fully operate in “Software as a Service” mode. This will support collaborative work, anyhow, it also has to cope with some threats and barriers occurring as a result of confidentiality requirements. In order to address the needs of more advanced users and their confidentiality, the need for a desktop-based interface was identified.

Table 11. Non-functional requirements.

Req. type	Requirements	Introduced
Usability	Diagnostic of input error	Semester 1
	Collaboration: model sharing	Semester 2
	Editor palette, properties, zoom	Semester 3
	Process oriented user interface	Semester 3
Ease of deployment	Web-based tool	Semester 1
	Desktop version packaging	Semester 4
Integration	Web-service API	Semester 2
	Early Warning System (Excel)	Semester 2
	Traceability of measures	Semester 2
	Interface for suppliers ratios (SAP, Oracle)	Semester 2
	Eclipse-based integration	Semester 4
Security	Project/User isolation	Semester 2
	Data confidentiality	Semester 3

4.3 Lessons Learned

Consistent User Experience Based on Process Support. An efficient way to provide a globally consistent user experience is to align the organisation of the user interface on the domain processes. In this case, this is particularly relevant because the risk management process shown in Fig. 1 is composed on a sequence of steps: risk identification, analysis, evaluation and treatment. Moreover, the process is iterative: after identifying measures, the model can be modified and then assessed again for the effectiveness of the measures or to decide about how to deal with residual risks. The way the process was reflected in the user interface is simply through a sequence of tabs that are followed one after the other: The first tab to build the supply chain models, second tab to decorated it with risks, and then to simulate and finally to analyse. This organisation is shown in the Fig. 12.

Hiding Modelling Complexity. Quantitative risk assessment needs to build a supply chain model with a precise semantics and capturing risks in mathematical form. In order to favor adoption by non specialists, especially in a SME context, those aspects should be hidden as much as possible. The following means were used to reach this goal:

- Graphical editor composed of supplier/process/storage nodes connected with flow of parts (either raw parts, partly assembled or final products). Such editor can easily be drawn by the end-user and automatically translated to the underlying simulation model (cf. Fig. 12).
- Risks as well as Key Performance Indicators can be quantified on the model using specific formulas referring to model elements. As those are evaluated

over specific model instance, we called them model queries. Writing such queries is not very complex but still requires some training to learn about the model primitive and how to combine them. In order to enable new users to assess some standard kinds of risk directly, a simplified risk model was made available to assess delay, quantity and quality risks. Through the use of a wizard depicted in Fig. 13, the user can directly instantiate the model without having to write any formula.

– In the same spirit, macros can also be developed for more specific problems, e.g. for dealing with sustainability. In this context, it is possible to provide estimators for process energy consumption or supplier CO_2 emission.

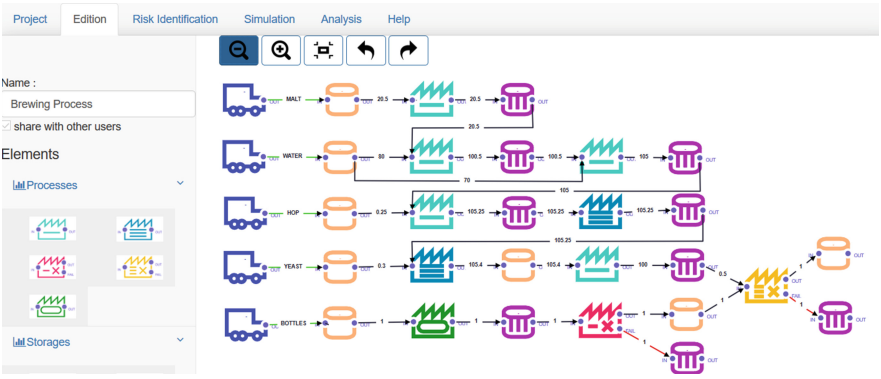


Fig. 12. Process-oriented user interface.

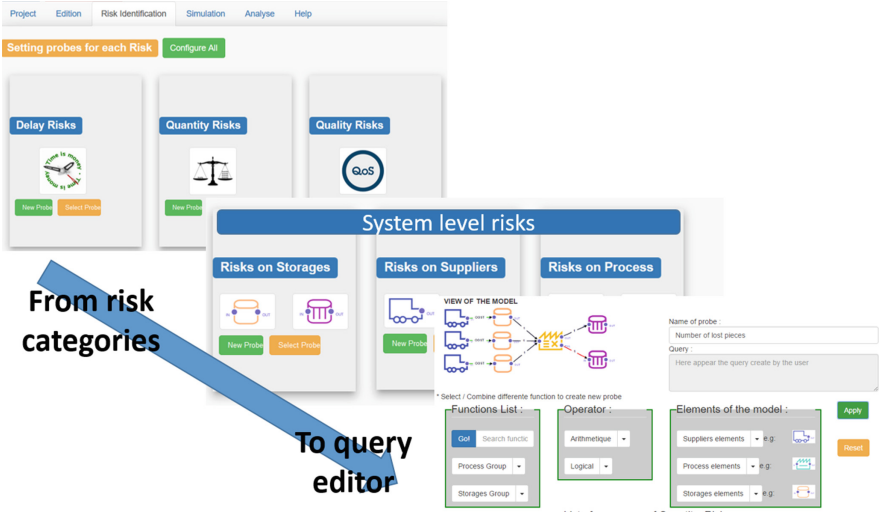


Fig. 13. Wizard for easing the assessment of standard risks.

Drawbacks of Web-Tools. While web tools operating in SaaS mode are attractive because it does only require a browser on the user side and limits the update on the server side. However, it has some drawbacks one must be aware of, such as:

- Development complexity can be non-trivial, especially when dealing with evolved user interfaces like graphical editors, dashboards and a wizard. This complexity can strongly reduce the ability to deliver new features.
- Browser compatibility is not always guaranteed. The used javascript framework (especially for graphical editors) should be tested with care.
- Security can also become a major barrier as companies can be reluctant to let their process and risk models be processed in servers outside their premises.

Those drawbacks motivated us to develop two complementary user interfaces connected to the same simulation engine:

- a web-based user interface simpler and used for awareness and for SME having no security concerns.
- an desktop (Eclipse-based) tool provided more advanced capabilities in terms of functionality, security and integration, but more complex to use.

Benefits of Open Tools. When developing features it is always useful to favour open design and open format that will enable the adopting user to further adapt the tool to its own need. The desktop version of the tool is based on the Eclipse Open Source platform and relies on highly configurable components. For example, the look-and-feel and even behaviour of the SIRIUS-based graphical editor can be tuned using an XML file [28]. In the same spirit, the report generation is based on BIRT that supports many output format (both Open Document and Microsoft format) and is based on templates that can easily be customised by end-users through a WYSIWYG editor [29]. Figure 14 shows the export and display of simulation results under the form of probability distribution for different kind of risks.

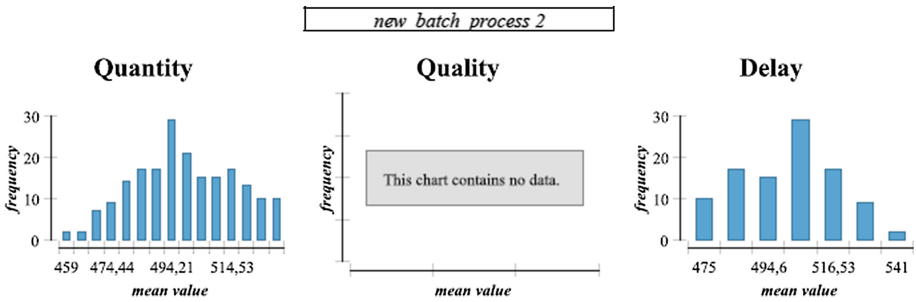


Fig. 14. Reporting using Open Source BIRT tool.

5 Conclusions

In this paper, we presented key requirements to build an efficient tool support for supply chains risk management. In order to best address the needs of SMEs, specific care was devoted to key adoption requirements like usability, learn-ability and integrability.

Rather than focusing on the presentation of the final requirements, we also detailed the process followed to gather them, based on a field survey and an Agile development process and summarised some lessons learned.

Despite some requirements could have been better anticipated, the flexibility of the Agile approach allowed us to do quick adjustments. In the end, we could produce two complementary user interfaces: a simpler web-based showcasing our tooling and a desktop based with more functionalities, integration and evolution capabilities. After successful validation, our work is now undergoing an open-sourcing process under the OscaR framework [30] and further rework for exploitation by SMEs.

Beyond the specific application field of supply chain, we also believe that the reported process and lessons learned can be useful for those engaging in a SME-oriented tool development in another field.

Acknowledgements. This research was conducted as part of the SimQRi research project (ERA-NET CORNET, Grant Nr. 1318172). The CORNET promotion plan of the Research Community for Management Cybernetics e.V. (IfU) is funded by the German Federation of Industrial Research Associations (AiF) based on an enactment of the German Bundestag.

References

1. Wieland, A., Wallenburg, C.M.: Dealing with supply chain risks: linking risk management practices and strategies to performance. Int. J. Phys. Distrib. Logistics Manag. **42**, 887–905 (2012)
2. Printz, S., von Cube, J.P., Vossen, R., Schmitt, R., Jeschke, S.: Ein kybernetisches modell beschaffungsinduzierter störgößen. In: Exploring Cybernetics - Kybernetik im interdisziplinren Diskurs. Springer Spektrum (2015)
3. Zsidisin, G.A., Ritchie, B.: Supply Chain Risk: A Handbook of Assessment, Management, and Performance. Springer, Boston (2009)
4. ISO: DIN ISO 31000: Risk management - Risk Assessment Techniques (2009)
5. Bernoulli, D.: Specimen theoriae novae de mensura sortis. Commentarii Academiae Scientiarum Imperialis Petropolitanae (1738)
6. Wells, G.: Hazard Identification and Risk Assessment. Institution of Chemical Engineers (1996)
7. Rausand, M.: Risk Assessment: Theory, Methods, and Applications. Statistics in Practice. Wiley, Hoboken (2011)
8. Artikis, C., Artikis, P.: Probability Distributions in Risk Management Operations. Springer, London (2015)
9. Zio, E.: The Monte Carlo Simulation Method for System Reliability and Risk Analysis. Springer, London (2013)

10. Gleißner, W.: Quantitative methods for risk management in the real estate development industry. J. Property Investment Finan. **30**(6), 612–630 (2012)
11. Deleris, L., Erhun, F.: Risk management in supply networks using Monte-Carlo simulation. In: 2005 Winter Simulation Conference, Orlando, USA (2005)
12. Chahal, K., Eldabi, T.: Which is more appropriate: a multi-perspective comparison between system dynamics and discrete event simulation. In: European and Mediterranean Conference on Information Systems (2008)
13. Finke, G.R., Schmitt, A., Singh, M.: Modeling and simulating supply chain schedule risk. In: 2010 Winter Simulation Conference, Baltimore, USA (2010)
14. Heckmann, I., Comes, T., Nickel, S.: A critical review on supply chain risk Definition, measure and modeling. Omega **52**, 119–132 (2015)
15. Blackhurst, J.V., Scheibe, K.P., Johnson, D.J.: Supplier risk assessment and monitoring for the automotive industry null. Int. J. Phys. Distrib. Logistics Manag. **38**, 143–165 (2008)
16. Chopra, S., Sodhi, M.S.: Managing risk to avoid supply-chain breakdown. MIT Sloan Manag. Rev. **46**, 53 (2004)
17. Mangla, S.K., Kumar, P., Barua, M.K.: Prioritizing the responses to manage risks in green supply chain: an Indian plastic manufacturer perspective. Sustain. Prod. Consumption **1**, 67–86 (2015)
18. Manuj, I., Mentzer, J.T.: Global supply chain risk management strategies. Int. J. Phys. Distr. Logistics Manag. **38**, 192–223 (2008)
19. Oke, A., Gopalakrishnan, M.: Managing disruptions in supply chains: a case study of a retail supply chain. Int. J. Prod. Econ. **118**, 168–174 (2009)
20. Punniyamoorthy, M., Thamaraiselvan, N., Manikandan, L.: Assessment of supply chain risk: scale development and validation. Benchmark. Int. J. **20**, 79–105 (2013)
21. Sodhi, M.S., Lee, S.: An analysis of sources of risk in the consumer electronics industry. J. Oper. Res. Soc. **58**, 1430–1439 (2007)
22. Sodhi, M.S., Tang, C.S.: Managing Supply Chain Risk. International Series in Operations Research & Management Science, vol. 172. Springer, Boston (2012)
23. Ziegenbein, A.: Supply Chain Risk Assessment: A Quantitative Approach. ETH-Zentrum für Unternehmenswissenschaften, Zürich (2006)
24. Printz, S., von Cube, J.P., Massonet, P.: SimQRi - simulative quantification of procurement induced risk consequences and treatment impact in complex process chains (2014). http://www.simqri.com
25. Printz, S., von Cube, J.P., Ponsard, C., De Landtsheer, R., Ospina, G., Massonet, P., Schmitt, R., Jeschke, S.: A survey on risk-management and tooling support for procurement processes in supply chains. In: Proceedings of the 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH) (2016)
26. Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J.: Agile software development methods - review and analysis. Technical report 478, VTT PUBLICATIONS (2002)
27. Sommerville, I.: Software Engineering. Pearson, Boston (2011)
28. Obeo: Sirius Designer (2016). http://www.obeodesigner.com/sirius
29. BIRT: Business Intelligence and Reporting Tool (2005). http://eclipse.org/birt
30. OscaR: OscaR: Scala in OR (2012). https://bitbucket.org/oscarlib/oscar

Making Network Solvers Globally Convergent

Tanja Clees, Igor Nikitin, and Lialia Nikitina(✉)

Fraunhofer Institute for Algorithms and Scientific Computing,
Schloss Birlinghoven, 53754 Sankt Augustin, Germany
{tanja.clees,igor.nikitin,lialia.nikitina}@scai.fraunhofer.de

Abstract. Stationary network problems are considered, unifying linear Kirchhoff equations and non-linear element equations, possessing a proper signature of the derivatives. The global non-degeneracy of the Jacobi matrix is proven, providing the applicability of globally convergent tracing algorithms for such systems. It is shown that stationary problems in gas transport networks can be written in the form necessary for the global convergence. Two stabilized algorithms for the solution of these problems are implemented. The algorithms outperform a standard Newtonian solver for a number of realistic networks. The algorithms do not depend on the starting point, are stable, converge for all scenarios and, additionally, provide feasibility indicator for the problem statement.

Keywords: Non-linear systems · Globally convergent methods · Stationary network problems

1 Introduction

This paper is an extension of our previous work [1], where we have began investigating a possibility to stabilize Newtonian solvers of the stationary network problems on example of gas transport networks and piecewise-linear Katzenelson algorithm. Here we add our new research results on generic non-linear Armijo stabilizer and compare both algorithms with the standard Newtonian solver on a number of realistic network scenarios. We also go deeper in details describing the modeling of gas transport networks, in particular, add a proof that all physically stable network elements also possess a signature necessary for stability of the solver.

Simulation generally requires to solve the large systems of non-linear equations. Normally Newtonian methods are applied for solution of such systems equipped with stabilizers such as backtracking line search. Usually to achieve convergence one should select a starting point in a basin of attraction of the method, sufficiently close to the solution. This makes the whole solution procedure a game of chance depending on the correct guess for the starting point. The situation is complicated by possible existence of multiple solutions or, even worse, the absence of solution. With the ordinary Newtonian method one cannot

M.S. Obaidat et al. (eds.), *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing 676,
https://doi.org/10.1007/978-3-319-69832-8_9

distinguish a failure of the solver from the real absence of solution (infeasibility) of the problem.

On the other hand, there are globally convergent Newtonian methods developed over the years. Whenever these methods are applicable, the solution exists and is unique, and convergence to it is guaranteed from an arbitrary starting point. The methods are applicable when the mapping $f: R^n \to R^n$ representing the system of equations $f(x) = 0$ has non-degenerate Jacobian in the whole space. In addition f must be proper, i.e., pre-image of every compact set is compact. In this case f becomes a diffeomorphism, i.e., is invertible and both f and f^{-1} are differentiable. This property guarantees the existence and uniqueness of the solution. The following tracing algorithm can be used to find the solution, see Fig. 1. One starts from an arbitrary point x_0, takes its image $y_0 = f(x_0)$, connects y_0 with the origin $y = 0$, e.g., by a straight line and reconstructs pre-image of this line in x-space. The obtained line in x-space goes from the starting point x_0 to the solution x^* we are looking for.

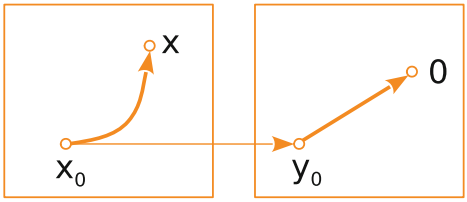


Fig. 1. Globally convergent tracing algorithm. Pre-image space (on the left) and image space (on the right) are linked by a diffeomorphism. One connects an image y_0 of a starting point x_0 with the origin $y = 0$ by a straight line and reconstructs its pre-image.

In this paper we will show how to apply these ideas to the solution of stationary problems in transport networks. We will use the property proven in our conference paper [1], that network problems, described by a combination of linear Kirchhoff equations and non-linear element equations of the form $f(P_{in}, P_{out}, Q) = 0$, possessing a signature of derivatives $\nabla f = (+ - -)$, have globally non-degenerate Jacobian. Considering gas transport networks, we show that the necessary signature is satisfied in a physical domain, i.e., in pressure range 1...150 bar. This means that the equations in the physical domain do not require any modification and whenever a solution in the physical domain exists, it is unique. Further one needs to continue the equations beyond the physical domain keeping the same signature of the derivatives. The solver occasionally performs iterations in the non-physical domain and there should not be any fold or other topological singularity hindering the convergence. So all the folds must be eliminated from the non-physical domain. After that the solution exists and is unique in the whole space. Whenever it belongs the non-physical domain, this is an indicator that the original system does not have a solution in the physical domain, i.e., is infeasible.

Globally non-degenerate Jacobian is not sufficient yet for the global convergence. One needs to use Newtonian stabilizers with theoretically guaranteed convergence. The necessary proofs are available for Armijo rule [2], Katzenelson algorithm [3] and their variants [4–7].

The present paper is organized as follows. In order to set the stage, in Sect. 2 we repeat the proof [1] of global non-degeneracy of Jacobian for transport network problems with a proper signature of non-linear elements. Then, in Sect. 3 we present suitable Newtonian stabilizers providing global convergence of the solver. In Sect. 4 we apply the methods to simulation of gas transport networks and discuss the obtained numerical results.

2 Generalized Resistive Systems

We will consider systems of the form [1]:

$$\sum_e I_{ne} Q_e = Q_n^{(s)},\ f_e(P_{in}, P_{out}, Q_e) = 0, \tag{1}$$

where indices $n = 1...N$ denote the nodes and $e = 1...E$ the edges of the network graph, I_{ne} is an incidence matrix of the graph, Q_e are flows through the edges, $Q_n^{(s)}$ are source/sink contributions, localized in supply/exit nodes, P_n are nodal variables (pressure, voltage, etc. – dependent on the context). The element equations possess derivatives of the signature:

$$\partial f_e/\partial P_{in} > 0,\ \partial f_e/\partial P_{out} < 0,\ \partial f_e/\partial Q_e < 0. \tag{2}$$

Lemma. System (1) under condition (2) possesses globally non-degenerate Jacobi matrix.

Proof. The Jacobi matrix of the system has the form:

$$J = \begin{pmatrix} 0 & I \\ \tilde{I}^T & -R \end{pmatrix}, \tag{3}$$

where $\tilde{I}$ contains derivatives $\partial f_e/\partial P_n$ of the element equations and R is a diagonal matrix containing positive entries $-\partial f_e/\partial Q_e$. Note that $\tilde{I}$ has the same pattern and the signs of entries as I. After elementary transformations we have

$$\det J = \prod_e (-R_e)\ \det I R^{-1} \tilde{I}^T. \tag{4}$$

The matrix $\tilde{L} = I R^{-1} \tilde{I}^T$ has sizes $N \times N$ and possesses a pattern and signs of entries identical with the Laplacian matrix $L = I\ I^T$. For the graphs containing several connected components both matrices have a block-diagonal structure, where we can select one block and further consider $L, \tilde{L}$ corresponding to one connected component. Both matrices have one zero eigenvalue, corresponding to an obvious eigenvector $v = (1...1)$. The difference between L and $\tilde{L}$ is that

L is symmetric and v is its left and right eigenvector, while $\tilde{L}$ is not symmetric and v is its left eigenvector. This degeneracy is related to the structure of the incidence matrix, i.e., every column of I contains only two entries $+1$ and -1, so that the sum of entries in every column vanishes.

Consider any other vectors annulating $\tilde{L}$ from the left:

$$(v\tilde{L})_{n'} = \sum_{n,e} v_n I_{ne} R_e^{-1} \tilde{I}_{n'e} = v_{n'} d_{n'} - \sum_{n \neq n'} v_n u_{nn'} = 0, \tag{5}$$

$$d_{n'} = \sum_e I_{n'e} R_e^{-1} \tilde{I}_{n'e},\ u_{nn'} = -\sum_e I_{ne} R_e^{-1} \tilde{I}_{n'e}.$$

Here we separate diagonal and non-diagonal contributions and see from the sign patterns of I and $\tilde{I}$ that $d_{n'} > 0$, while $u_{nn'} > 0$ if $n \neq n'$ are connected by an edge and $u_{nn'} = 0$ otherwise. Also, from the annulation of the matrix by $v = (1...1)$ we have

$$d_{n'} = \sum_{n \neq n'} u_{nn'}. \tag{6}$$

Thus we have

$$\sum_{n \neq n'} (v_{n'} - v_n) u_{nn'} = 0. \tag{7}$$

Let us consider a maximal entry $v_{n'} \geq v_n$ for all $n \neq n'$. The above condition gets LCP form, satisfied only if

$$(v_{n'} - v_n) u_{nn'} = 0, \tag{8}$$

i.e., for connected nodes $v_{n'} = v_n$ must be satisfied. In this way the maximal value propagates to all connected nodes in the graph, leading to the conclusion that $v_n = Const$ is the only solution. Therefore, only the vectors proportional to $v = (1...1)$ are the left annulators of $\tilde{L}$.

To eliminate the degeneracy, one has to remove (at least) one Kirchhoff equation in the connected component and fix the corresponding nodal variable to a constant value. Physically this corresponds to the creation of an entry point for the flow and setting a pressure (voltage, etc.) value there: $P_n = Const$ for $n \in Pset$. On matrix level it corresponds to the removal of (at least) one row from the matrices $I, \tilde{I}$ and the corresponding row and column from $L, \tilde{L}$. Searching the left annulators of $\tilde{L}$, we obtain a similar system but with $v_n = 0$ for $n \in Pset$ and the equations with $n' \notin Pset$. We come to the conclusion that the maximal value $v_{n'}$ propagates to all connected nodes $\notin Pset$ and to their neighbors $\in Pset$, where $v_n = 0$ is set. Thus, the maximal value $v_{n'} = 0$. Similarly, considering the propagation of the minimal value, we also have $v_{n'} = 0$. Therefore, $v = 0$ is the only solution, the matrix $\tilde{L}$ does not have non-zero annulators and $\det \tilde{L} \neq 0$.

Further, it is easy to show that $\det \tilde{L} > 0$. The standard Laplacian matrix L is symmetric and positive semi-definite. Removing the $Pset$ nodes makes it strictly positive definite, so that $\det L > 0$. Considering a linear homotopy

$$\begin{aligned} &\hat{I}(\lambda) = I(1-\lambda) + \tilde{I}\lambda,\ \hat{R}(\lambda) = 1 \cdot (1-\lambda) + R\lambda, \\ &\hat{L}(\lambda) = I\hat{R}^{-1}(\lambda)\hat{I}^{\,T}(\lambda),\ \hat{L}(0) = L,\ \hat{L}(1) = \tilde{L}, \end{aligned} \tag{9}$$

during the transition $\lambda \in [0,1]$ all matrices have the same sign pattern as at the beginning and at the end of the path. We know that $\det L > 0$. If $\det \tilde{L} < 0$, we can find a point in between where $\det \hat{L}(\lambda) = 0$, i.e., find a degenerate matrix in the considered class of matrices, which, as we have just proven, does not exist. Thus, $\det \tilde{L} > 0$.

Finally, we conclude that $\det J$ does not vanish and has a sign $(-1)^E$ defined only by the number of edges in the connected component. In particular, when piecewise linear element equations are considered, $\det J$ has the same sign in all pieces. ■

The systems (1) under condition (2) will be further called generalized resistive systems. The "purely" resistive systems, depending on the difference of nodal variables, form a special subclass of generalized resistive systems, whose element equations can depend on nodal variables separately. This special subclass corresponds to $\tilde{I} = I$ and a symmetric Jacobi matrix. Our main conclusion is that the generalized resistive systems can be treated in pretty the same way as resistive ones, although their Jacobi matrix is not symmetric anymore.

3 Globally Convergent Algorithms

Global non-degeneracy of Jacobian is not yet sufficient for global convergence of the solver. In practice, accelerated algorithms are often used, e.g., [8,9], for which the global convergence is not always guaranteed. It is important to choose the algorithms with theoretically proven convergence, to make use of the advantages of globally non-degenerate Jacobian. In particular, one should not allow Newtonian iteration to take always the full step. Every Newtonian step dx is a solution of linearized problem $J(x)dx = -f(x)$, where the linearization becomes less and less applicable when the step length is increased. Without a control of the step Newtonian iterations would go to infinity or create complex and beautiful fractal structures, practically meaning divergence of the method. On the other hand, too small steps lead to a slowdown. A stabilization algorithm should provide a good balance between the restriction of the step and convergence rate of the method.

Armijo stabilizer. The algorithm from [2] is one of the standard stabilizers and is well described in the textbooks, e.g., in [6]. It works for generic non-linear problems with globally non-degenerate Jacobian. The idea is to restrict the Newtonian step by observing a merit function, usually a norm of the residuals $||f(x)||$. The necessary condition is a simple decrease of the merit function

$||f(x+\lambda dx)|| < ||f(x)||$, while to avoid the algorithm of getting stuck, a condition of sufficient decrease is formulated: $||f(x+\lambda dx)|| < (1-\alpha\lambda)||f(x)||$ with a constant $\alpha \in (0,1)$.

Algorithm (backtracking line search):

```
repeat until convergence:
   do Newtonian step dx
   set λ = 1
   trial point: x_t = x + λdx
   do λ = λ/2                                          (* bisection *)
   until ||f(x_t)|| < (1 − αλ)||f(x)||     (* sufficient decrease of residual *)
   x = x_t                                          (* trial point accepted *)
```

The algorithm is globally convergent, provided that $J(x)$ is Lipschitz continuous and $||J^{-1}(x)|| < C$. The proof can be found in [6].

Katzenelson stabilizer. The algorithm has been originally formulated in paper [3] and works for continuous piecewise linear resistive systems. Such systems are composed of linear Kirchhoff equations and piecewise linear element equations of the form $f(P_{in}-P_{out}, Q) = 0$, relating a difference of nodal variables $P_{in}-P_{out}$ to the flow Q through the element. Continuity means that the mapping $y = f(x)$ as a whole is C^0-continuous, resistiveness means that the element equation in every piece can be written as $P_{in} - P_{out} = RQ + Const$, where $R > 0$ is the resistance. As a result, the derivatives of the element equation possess the signature (2). Under these conditions the Jacobian determinant $\det J$ of the system has the same sign in all pieces. Although this mapping is not a diffeomorphism anymore, it is a homeomorphism, every point y still has a single pre-image x. The linear system in every piece essentially coincides with the one appearing in Newtonian step. The only necessary modification is that, whenever the step attempts to leave the piece, the algorithm should stop at the border:

Algorithm (piecewise linear tracing):

```
do Newtonian step using the current Jacobi matrix;
if the solution leaves the piece:
   stop at the border;
   update the Jacobi matrix to the other side;
   proceed to the next piece;
else:
   return the solution.
```

The algorithm is globally convergent and comes to the solution in a finite number of steps. The proof is given in [4]. This paper also relaxes the condition that $\det J$ has to be of the same sign in all pieces. This condition has been imposed only on unbounded pieces, while in bounded ones $\det J$ can change sign and even vanish, producing a locally degenerate system. The algorithm has been extended to process such cases, still providing convergence in a finite number of steps.

In paper [5] the continuous piecewise linear mappings were represented in a so called *max-min form*:

$$f(x) = \max_i \min_j a_{ij}^T x + b_{ij}. \tag{10}$$

Further, using for max-min functions their definition in terms of absolute values:

$$\begin{aligned} \max(x, y) &= (x + y + |x - y|)/2, \\ \min(x, y) &= (x + y - |x - y|)/2, \end{aligned} \tag{11}$$

continuous piecewise linear systems have been transformed to systems combining globally linear functions and absolute value functions. The paper discusses the relation of such systems with linear complementarity problems (LCPs) and Karush-Kuhn-Tucker (KKT) conditions and presents a number of algorithms for their solution, converging in a finite number of steps.

Strictly speaking, algorithms from Katzenelson family are designed for piecewise linear systems, however, in practice, they are also applicable for the systems composed of piecewise linear and (slightly) non-linear parts, which particularly appear in gas transport networks. The reason is that stops at the borders still work as a good stabilizer to Newtonian method. Katzenelson algorithm is also less sensitive to instabilities appearing for almost degenerate systems, e.g., when some of the resistances tend to zero or infinity. On the other hand, Armijo algorithm is applicable for generic non-linear systems. In certain cases it can overperform Katzenelson algorithm, jumping over multiple boarders at once, if the merit function will allow this jump.

4 Application to Gas Transport Networks

In addition to pressure P, gas transport networks possess several other nodal variables: density ρ, compression factor z, absolute temperature T, and further characteristics, dependent on gas composition, such as molar mass μ, critical pressure P_c, critical temperature T_c, etc. The gas characteristics are distributed over the network by a principle of molar mixing. These values enter into equation of state (EOS), which can be written in a form $P = f(\rho, ...)$ or $\rho = f^{-1}(P, ...)$.

An important property for stability of the network is that function f monotonously increases w.r.t. the first argument. Indeed, consider a gas volume, depicted on Fig. 2 left, in equilibrium with external pressure P_{ext}. Suppose that diameter of the volume is decreased as a result of a fluctuation. This leads to increase of the density ρ. If EOS provides increasing internal pressure P, collapse of the volume will be stopped. This behavior forms a typical negative feedback loop stabilizing the diameter of the volume. Otherwise, if EOS provides decreasing pressure, the feedback is positive and collapse will continue until the volume shrinks into a point. We see that stability of the medium leaves only EOS with the signature of the first type.

Further, consider an element, depicted on Fig. 2 right, described by a certain equation of the element (EOE). The element is equipped with two reservoirs and

a stationary solution with a fixed throughput Q_{set} is considered. Let, at first, the input pressure be fixed: $P_{in} = P_{set}$. Suppose that the output pressure P_{out} is decreased as a result of a fluctuation. If EOE will provide the increase of the flow Q, then the gas will be accumulated in the output reservoir and increase its density ρ_{out}. EOS will provide the increase of the output pressure. This forms a negative feedback loop, returning the element to the original state. Otherwise, if EOE will provide decreasing Q, the output pressure will continue to drop. So again, the positive feedback will render the element unstable. Now let the output pressure be fixed: $P_{out} = P_{set}$. Suppose that the input pressure P_{in} is decreased as a result of a fluctuation. If EOE will provide the decrease of the flow Q, then the gas will be accumulated in the input reservoir and the input pressure will increase. The negative feedback indicates stability of the element. Otherwise, if EOE will provide increasing Q, the input pressure will fall, the positive feedback indicates instability of the element.

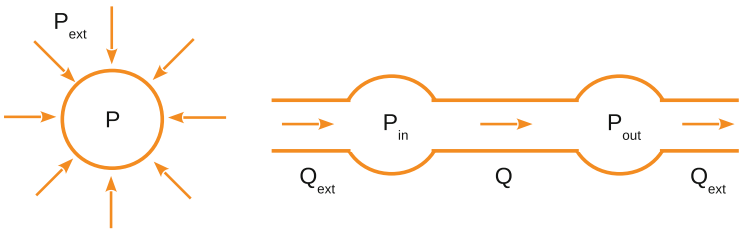


Fig. 2. Analysis of stability in gas transport network: on the left – stability of the physical medium (EOS), on the right – stability of the flow through the element (EOE).

In summary, this analysis is shown in Table 1. The negative feedback is available only if EOE provides the change of Q opposite to the change of P_{out} at fixed P_{in} and the change of Q in the same direction as the change of P_{in} at the fixed P_{out}, as it should be for the signature $\nabla f = (+--)$. As a result, we have proven that physical stability of the elements in gas transport network implies the generalized resistive condition.

Table 1. Analysis of the physical stability of an abstract element in the gas network.

Fixed	Fluctuation	EOE	EOS	Feedback
P_{in}	$P_{out} \downarrow$	$Q \uparrow$	$P_{out} \uparrow$	$-$
P_{in}	$P_{out} \downarrow$	$Q \downarrow$	$P_{out} \downarrow$	$+$
P_{out}	$P_{in} \downarrow$	$Q \downarrow$	$P_{in} \uparrow$	$-$
P_{out}	$P_{in} \downarrow$	$Q \uparrow$	$P_{in} \downarrow$	$+$

Continuation. In practice, EOS and EOE are given by approximate formulae valid in a certain domain, e.g., for gas transport networks pressure range

1...150 bar and flows 0...1000 Nm^3/h. The physical stability of the elements implies correct signature condition inside this domain, while outside the equations can be continued using the following universal formula:

$$\begin{aligned} f(x_1,...,x_n) &= f(\hat{x}_1,...,\hat{x}_n) \\ &+ \sum_{k=1}^{n} (\min(x_k - a_k, 0) + \max(x_k - b_k, 0)), \\ \hat{x}_k &= \min(\max(x_k, a_k), b_k). \end{aligned} \tag{12}$$

The formula describes a continuation of a function of n variables, specified in a box $a_k \le x_k \le b_k$, to the whole R^n. The function in the box is monotonously increasing w.r.t. each argument and its continuation also possesses this property. The first term in this formula clamps the arguments into the box, while the further terms provide correct signature of the gradient outside of the box. The connection of the functions on the border of the box is C^0-continuous. Better smoothness can be achieved as follows. The max-min functions can be transformed to an absolute value representation (11), then one can use a smooth regularization for the absolute value function, e.g.,

$$|x|_\epsilon = \sqrt{x^2 + \epsilon^2}. \tag{13}$$

This substitution improves the smoothness of the connection, actually making it C^∞-continuous.

In our application, the continuation formula (12) with $n = 1$ can be used directly for monotonously increasing function $\rho = f(P, ...)$ in EOS. For EOE with a signature $\nabla f = (+ - -)$ one can apply a suitable replacement $f(P_{in}, -P_{out}, -Q)$, making the function monotonously increasing w.r.t. each argument, then use the continuation formula (12) with $n = 3$.

Elements. Further we describe the implementation details for simulation of gas transport networks, performed by the software Mynts (Multi-phYsics NeTwork Simulator [10]), developed in our group. For the experiments we used several test networks of various complexity, from the simplest 100-nodal one, shown on Fig. 3, to 4K-nodal real-life network. In Table 2 the detailed parameters of the test networks are given. Figure 4 shows the internal structure of compressor and regulator stations in the networks. Typical elements used in gas transport networks are listed below.

Pipes: in the simplest case can be represented by a quadratic resistive model [11]

$$P_{in}|P_{in}| - P_{out}|P_{out}| = RQ|Q|, \tag{14}$$

where P is a pressure, Q is a flow, R is a resistance coefficient, depending on diameter, length and friction characteristics of the pipe. For realistic modeling more complicated element equations can be used, based on friction models by Nikuradze, Hofer or Prandtl-Colebrook [11,12].

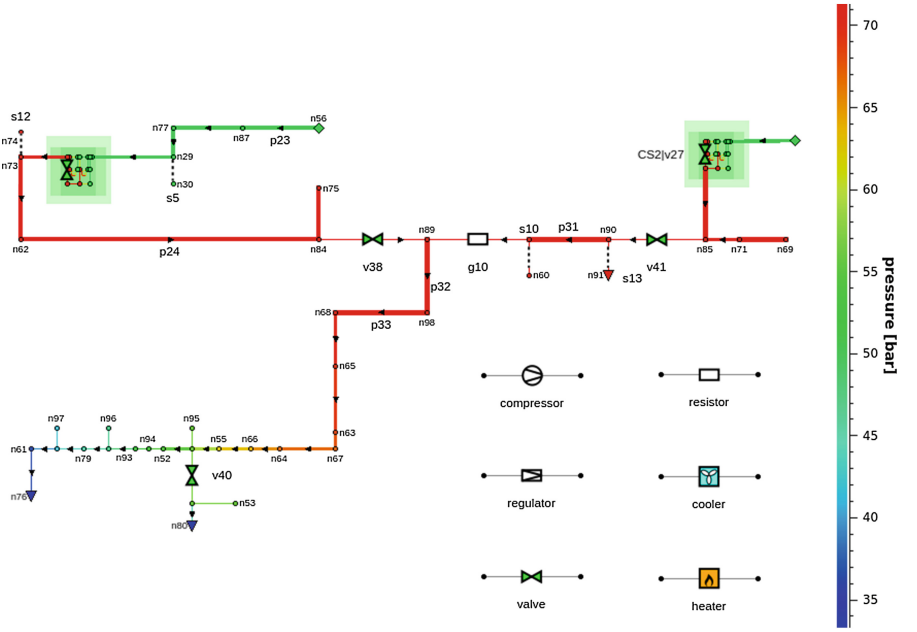


Fig. 3. Gas transport network simulation in Mynts. The network topology with the resulting pressure distribution, shown by color.

Resistors: physically correspond to short pipe segments, described by the same quadratic formula, with R specified explicitly.

Valves: simple switching elements

$$P_{in} = P_{out} \text{ (open)}; \quad Q = 0 \text{ (closed)}. \tag{15}$$

Regulators: control elements dropping pressure, physically correspond to a variable resistor, whose value is automatically adjusted to satisfy one of the following control goals: a fixed output pressure (SPO), a fixed input pressure (SPI) or a fixed flow value (SM). Being combined with the given upper and lower bounds: $PH = \min(SPO, POMAX)$, $PL = \max(SPI, PIMIN)$, $QH = \min(SM, MMAX)$, the element equation defines a polyhedral surface shown in Fig. 4, top. Here every face corresponds to the best possible satisfaction of the control goal, e.g., $P_{out} = PH$ (typical for SPO-mode), $Q = QH$ (typical for SM-mode), $P_{in} = P_{out}$ (bypass BP, equivalent to an open valve), $Q = 0$ (OFF, equivalent to a closed valve), etc. Like any piecewise linear equation, it can be represented in max-min form:

$$\begin{aligned} &\max(\min(\min(\min(P_{in} - PL, -P_{out} + PH),\\ &-Q + QH), P_{in} - P_{out}), -Q) = 0. \end{aligned} \tag{16}$$

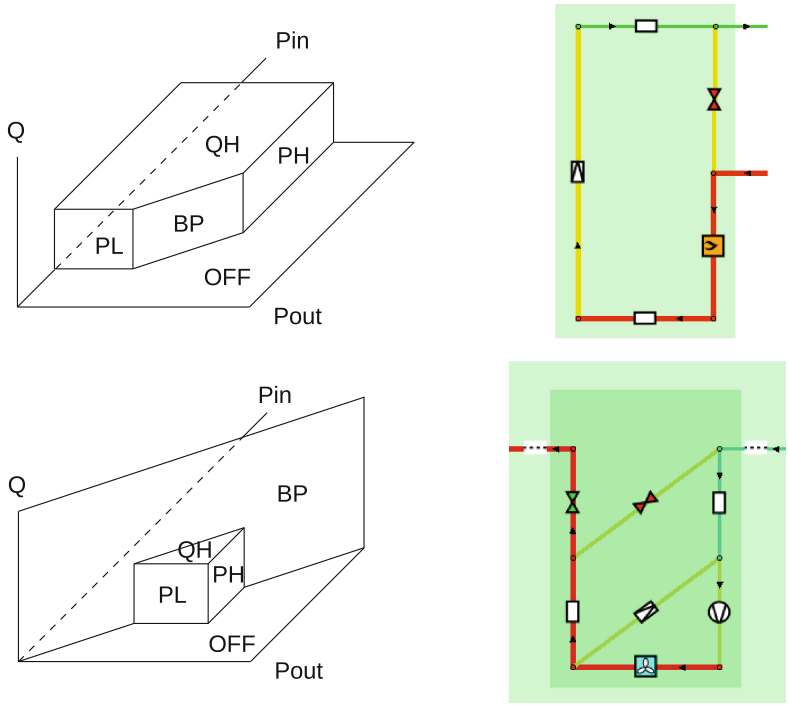


Fig. 4. Gas transport network simulation in Mynts (cont'd). On the top – a regulator station, on the bottom – a compressor station. On the left – control element diagrams, on the right – internal structure of the stations.

Compressors: control elements increasing pressure, they also have a resistive equation, similar to a battery with internal resistance in electrotechnics. They also can be configured to satisfy the control goals of SPO, SPI and SM type. The element equation defines a polyhedral surface shown in Fig. 4, bottom. The corresponding max-min form is:

$$\begin{aligned}&\max(\max(\min(\min(P_{in}-PL,-P_{out}+PH),\\&-Q+QH),P_{in}-P_{out}),-Q)=0.\end{aligned}\tag{17}$$

Nodal equations: have the form of EOS

$$P=f(\rho,T,...)\text{ or }\rho=f^{-1}(P,T,...),\tag{18}$$

using an approximation formula [12,13], such as AGA, Papay, ISO standard AGA8-DC92, etc. As we have already shown, the physical stability of the medium requires that the functions f, f^{-1} increase monotonously w.r.t. the first argument, in this way supporting the existence and uniqueness of a solution.

In real scenarios further variables and equations are added, describing temperature distribution, gas composition, more detailed modeling of control elements, etc.

Regularization. To provide strict resistive property for all elements, their equations should be properly regularized. In pipe/resistor equations one needs to add a laminar term ϵQ to the right hand side, where ϵ is a small positive constant. This term provides non-degeneracy of the system and correct signature of the equation for $Q = 0$. Also the $\epsilon(P_{in} - P_{out})$ term should be added to the left hand side to protect the system from similar problems at $P = 0$. Note that the absolute value function in the $Q|Q|$ and $P|P|$ terms has a different meaning: $Q|Q|$ provides the correct symmetry of the equation in reversal of the flow direction, while $P|P|$ removes a fold in the mapping and provides the existence and uniqueness of a solution everywhere in the space of variables, including the non-physical domain $P < 0$. As we already mentioned, the physical solution cannot be located in this domain, however, the tracing algorithm can wander there on intermediate iterations.

For valves and control elements one should also introduce regularization terms to provide the resistive signature for every face of the element equation. Properly regularized equations have the form:

Valves:

$$\begin{aligned} &P_{in} - P_{out} = \epsilon Q \quad \text{(open)};\\ &Q = \epsilon(P_{in} - P_{out}) \quad \text{(closed)}. \end{aligned} \tag{19}$$

Regulators:

$$\begin{aligned} &\max(\min(\min(\min(P_{in} - \epsilon P_{out} - \epsilon Q - PL,\\ &\epsilon P_{in} - P_{out} - \epsilon Q + PH), \epsilon(P_{in} - P_{out}) - Q + QH),\\ &P_{in} - P_{out} - \epsilon Q), \epsilon(P_{in} - P_{out}) - Q) = 0. \end{aligned} \tag{20}$$

Compressors:

$$\begin{aligned} &\max(\max(\min(\min(P_{in} - \epsilon P_{out} - \epsilon Q - PL,\\ &\epsilon P_{in} - P_{out} - \epsilon Q + PH), \epsilon(P_{in} - P_{out}) - Q + QH),\\ &P_{in} - P_{out} - \epsilon Q), \epsilon(P_{in} - P_{out}) - Q) = 0. \end{aligned} \tag{21}$$

Feasibility Indicator. The obtained system belongs to the generalized resistive type and, therefore, it always has a unique solution. On the other hand, it can happen in real scenarios that they do not have a solution. The determination whether a solution for given conditions exists represents a so-called feasibility problem. Usually solutions disappear when one requires too much from the network, e.g., to transport a large amount of gas through a long pipe system with only one supply where Pset = 10 bar and all compressors are switched off. There is no physical solution for such a scenario, while a solution of our generalized resistive system will exist. This solution, however, will be located in the non-physical domain, where some nodes have negative pressure. This can be used as an indicator of feasibility for the tested scenario.

Practically, observing the work of the algorithm, we often see that a solution goes to the non-physical domain, wandering there along complex trajectories. Finally it either returns to the physical domain if the problem is feasible or remains in the non-physical domain otherwise. Considering the solution as the function of a regularization parameter $x^*(\epsilon)$ and removing the regularization $\epsilon \to +0$, we observe that the solution for feasible problems will have a limit in the physical domain, while for infeasible ones it either has a limit in the non-physical domain or tends to infinity.

We note that ϵ-regularization is one possibility to provide global convergence of the tracing algorithm. The other possibility is a modification of the algorithm described in [4], making it applicable also for degenerate Jacobi matrices encountered in bounded pieces. An investigation of this possibility is part of our further plans.

We have implemented two globally convergent algorithms in a test mode in our network simulator Mynts under the option solver_strategy = stable. Using a number of realistic scenarios from our partners, we have compared the performance of the algorithms vs. the option solver_strategy = standard, representing a generic Newtonian solver. The results of our comparison are presented in Table 2. We see that the generic solver provides worse convergence and diverges in certain scenarios, while the new algorithms always converge, in agreement with their theoretical properties.

Table 2. Gas transport network simulation, comparison of the algorithms. For every network two scenarios are considered, different by numerical values of $Pset$, $Qset$ and compressor/regulator SM, SPO settings. Divergent cases are marked as 'div'. Number of iterations (iter.) and timing (t) are given. Simulation is performed on a 3 GHz Intel i7 CPU 8 GB RAM workstation.

Network	Nodes	Edges	Scenario	Solver_strategy						Feasible?
				Standard		Stable				
						Armijo		Katzenelson		
				iter	t, s	iter	t, s	iter	t, s	
N1	100	111	S1	3	0.01	2	0.01	2	0.01	Y
			S2	57	0.17	11	0.03	4	0.02	Y
N2	931	1047	S1	11	0.27	12	0.31	8	0.25	Y
			S2	div	–	13	0.36	32	0.77	N
N3	4466	5362	S1	div	–	26	3.3	13	2.0	Y
			S2	47	6.5	26	3.3	14	1.9	Y

5 Conclusions

Stationary network problems have been considered, unifying linear Kirchhoff equations and non-linear element equations of the form $f(P_{in}, P_{out}, Q) = 0$, possessing a signature $\nabla f = (+ - -)$. The global non-degeneracy of the Jacobi

matrix has been proven, providing the applicability of globally convergent tracing algorithms for such systems. It has been shown that the stationary problems in gas transport networks can be written in the form necessary for the global convergence. Two stabilization algorithms have been implemented, Armijo backtracking line search and Katzenelson piecewise linear tracing. The algorithms have been applied to several realistic networks and their performance has been compared with a generic Newtonian solver. The generic solver provides slower convergence and fails in certain scenarios. The tracing algorithms have better performance and converge for all scenarios.

As a future work, we plan to enhance our approach with advanced modeling of gas compressor stations, including calibrated characteristics and measured physical profiles into the control element equations.

Acknowledgements. We are grateful to the participants of SIMULTECH 2016 conference for fruitful discussions.

References

1. Clees, T., et al.: A globally convergent method for generalized resistive systems and its application to stationary problems in gas transport networks. In: Proceedings of SIMULTECH 2016, pp. 64–70. SCITEPRESS (2016)
2. Armijo, L.: Minimization of functions having Lipschitz continuous first partial derivatives. Pac. J. Math. **16**(1), 1–3 (1966)
3. Katzenelson, J.: An algorithm for solving nonlinear resistor networks. Bell Syst. Tech. J. **44**(8), 1605–1620 (1965)
4. Chien, M.J., Kuh, E.S.: Solving piecewise-linear equations for resistive networks. Int. J. Circ. Theory Appl. **4**(1), 1–24 (1976)
5. Griewank, A., et al.: Solving piecewise linear systems in abs-normal form. Linear Algebra Appl. **471**, 500–530 (2015)
6. Kelley, C.T.: Iterative Methods for Linear and Nonlinear Equations. SIAM, Philadelphia (1995)
7. Allgower, E.L., Georg, K.: Introduction to Numerical Continuation Methods. SIAM, Philadelphia (2003)
8. Press, W.H., et al.: Numerical Recipes in C. Cambridge University Press, New York (1992)
9. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. **106**(1), 25–57 (2006)
10. Clees, T., et al.: MYNTS: multi-physics network simulator. In: Proceedings of SIMULTECH 2016, pp. 179–186. SCITEPRESS (2016)
11. Mischner, J., et al.: Systemplanerische Grundlagen der Gasversorgung. Oldenbourg Industrieverlag GmbH, München (2011)
12. Schmidt, M., et al.: High detail stationary optimization models for gas networks. Optim. Eng. **16**(1), 131–164 (2015)
13. DIN EN ISO 12213-2: Natural Gas – Calculation of compression factor, European Committee for Standartization (2010)

Predictive-Delay Control for Overloading in Real-Time Scheduling

Zakaria Sahraoui[1(✉)], Abdenour Labed[1], Mohamed Ahmed-Nacer[2], and Emmanuel Grolleau[3]

[1] Computer Science Department, École Militaire Polytechnique, BP 17, Bordj-Elbahri, Algiers, Algeria
z.sahraoui@gmail.com, abd.labed@gmail.com

[2] Computer Science Department, Université des Sciences et de la Technologie Houari Boumediene, Algiers, Algeria
anacer@mail.cerist.dz

[3] LIAS, ENSMA, Téléport 2, 1, Av. Clément Ader, BP 40109, 86961 Chasseneuil Futuroscope, Cedex, France
grolleau@ensma.fr

Abstract. In multi-task control systems, control quality is subject to deterioration due to system dynamics and several scheduling artefacts. Based on open-loop scheduler or feedback scheduling, we investigate the advantage of some new techniques, such as the subtask scheduling and the predictive delay control, used to deal with control and scheduling co-design constraints. In the latter, at each time instant, the measurement signal is predicted by extrapolation that minimizes the effect of the measurement obsolescence. This predictive method, compared to other complex dynamic methods, is easier to formulate and its results within a discrete-time control algorithm are suitable for embedded systems. In the present work, simulations are conducted to show that the predictive-delay control can improve the control quality even in the absence of a dynamic priority assignment like in the Earlies deadline First Algorithm. However, in order to take advantage from the potential of both methods namely the predictive-delay control and the subtask-scheduling, another alternative is to combine them in the same solution to better deal with the input-output latency. The experimental validation is accomplished using the servo-motor and the inverted-pendulum systems through a stochastic execution-time implementation.

Keywords: Overload · Predictive-delay · Input-output latency · Subtask scheduling · Real-time · Control · Performance analysis · Feedback-scheduling · TrueTime

1 Introduction

In embedded systems, control tasks design is often accompanied by other system tasks, with more or less hard real-time constraints (ex., decision, measurement,

M.S. Obaidat et al. (eds.), *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing 676,
https://doi.org/10.1007/978-3-319-69832-8_10

saving and communication tasks). The first characteristic of a control task is its recurrent or periodic nature. On one hand, we must choose the appropriate scheduler in order to respect real-time constraints for each task. But on the other hand, prior to the scheduler choice, one should determine a real-time executive, up to the application design (w.r.t. required services), that agrees with the associated funding. Whereas, commercial executive does not propose refined scheduling algorithms such as the earliest deadline first (EDF). This lack lets the way open for researchers to seek suitable solutions, and even if these algorithms are implemented, it is therefore essential to fill them in with all due care and attention.

In control theory, selection of appropriate task period is a fundamental constraint, while in scheduling theory, the processor load is one of the most common constraints. The choice of a processor for an embedded system is initially based on these two constraints, which means that there is a relationship between the period and the processor load. Furthermore, insuring schedulability does not necessarily mean control with high performance, and reducing the task periods does not necessarily increase the quality of control [1].

Open-loop algorithms schedulers such as the Rate or Deadline Monotonic (fixed priority) and EDF (dynamic priority) algorithms cannot manage the tasks system in processor overload and tasks overrun (i.e., missed deadlines) situations. They do not deal with the problem of period selection or takes care of the scheduling latencies. Several solutions have been proposed for the problem of period selection and the other scheduling and control co-design constraints. Feedback and feed-forward real-time mechanisms are used in [2–5] to handle the period rescaling problem. Nevertheless, the subtask scheduling [6], the control-server [2] and the predictive-delay control [1] deal with the input-output latency, which is a significant artifact that may deteriorate the control.

The real-time communities have been working on this subject for 20 years. The seminal work presented in [7], solves an optimization problem based on a non linear criterion, then in [8] other criteria are proposed for the optimization of control performance as a function of the period and the computing latency. Later, there has been suggestions to resolve other optimization problems on-line to fit the scheduling constraints as schedulability or task periods selection, like in [9] by RST & H∞ algorithms together or by the LPV method [3,4]. These solutions are referred to as the indirect feedback scheduling (FBS). Methods that suggest priority assignment, like in [5] with the LQG method or in [10–12] are called direct FBS. In the class of the direct FBS we also find the solution of [13,14] based on the Predictive Control Model.

Particularly, authors in [2,6] have studied the impact of the scheduling jitters on the QC using the jitterbug tool [2] and then those of the latencies on the QC using the TrueTime tool [15]. The authors, proposed an indirect FBS to rescale tasks periods, based on a processor load estimator. The subtask solution is reconsidered by [16,17] in order to enhance the schedulability under fixed priority scheduling or by [18–20] to minimize the input-output jitter. Finally, in [2] the subtask scheduling is used to improve the QC.

These methods are recent theories and research results which may be of valuable help. In this context, we aim through the present contribution to evaluate the predictive-delay control and the subtask scheduling methods, based or not on the feedback scheduling technique, using a static and a dynamic algorithms such as the RM and EDF schedulers. Firstly, in Sects. 2, 3 and 4, the task model, the processes, the experimental setting and scheduling techniques are detailed. In Sect. 5 an example, showing the benefits and backwards of all the above-mentioned methods, is presented.

In Sect. 6, using two case studies and intensive simulation where computing duration varies, we first show that the used FBS fails in stabilizing processes controlled by low priority tasks, in case of processor overload. Then, the subtask scheduling method studied in [2] and the Predictive-Delay Control (P-DC) proposed in [1] are tested, taking into account the effect of delays as an inherent characteristic of a feedback scheduling. Finally, after this analysis and comparison, we show that combining both of these methods leads to even better result.

2 Task Model and Experimental Settings

Lets first introduce the classical task model with the associated notation: we call tasks system the set of tasks $S = \{\tau_1, ...\tau_N\}$ involved in a given real-time system and denote the number of tasks by N. In addition, two jobs of a task are considered perfectly interchangeable in that they perform identical treatment.

A given task τ_i is characterized by its period h_i, its observed execution-time $C_i(k)$ at time index k, its worst execution time C_i and the date of its first arrival (or offset) O_i. The tasks systems studied in this work have implicit-deadlines (i.e., tasks must terminate before their next release). Each periodic task generates a potentially infinite set of jobs $\tau_i(k)$, where k refers to the k^{th} sampling period: every sub-request job is released every h_i time unit.

2.1 Task Division into Calculate-Output and Update-State

A typical model to get the minimum latency from the measure input to the control output is to split the controller code into two segments: Calculate-Output and Update-State. The control output is send to the process before the Update-State segment [21], see Listing 1. We implement the P-DC with this model for two reasons:

(i) to conform, in terms of matching and comparison, the P-DC method with the subtask scheduling which is based on this typical model,
(ii) to check the efficiency of this method with the minimum of latencies not due to scheduling artifacts.

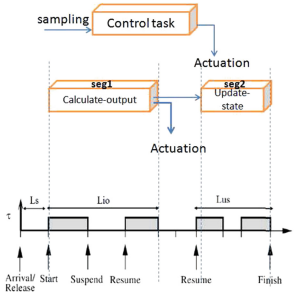


Fig. 1. Task division into Calculate-Output and Update-State.

In Fig. 1 the execution time of the Calculate-Output segment C_{co} is a rate of $C_i(k)$ (in %). This means that the delay from the jobs start time to the end of the Calculate-Output segment will be at least $C_{co}\, C_i(\mathrm{k})$. However, preemption from higher priority tasks may induce a longer delay, where the time from the jobs release/arrival time until its start time is noted the sampling latency Ls and Lio is the Input-Output latency representing the C_{co} segment latency. The second segment returns $C_{us}(\%)$ of $C_i(k)$, which is reserved to update the PID state variables. This duration can be also subject of preemption from higher priority tasks and noted by Lus as an Update-State Latency. Finally the response time latency is defined by

$$\mathrm{Lresp} = \mathrm{Ls} + \mathrm{Lio} + \mathrm{Lus}.$$

It is important to know that in the P-DC method the task scheduling is assumed by the RM scheduler under the FBS. Nevertheless, in subtask scheduling, we assign the priorities to the tasks segments (subtask model) where the scheduling is assumed with the FP protocol. This technique is proposed in [2] and implemented under the TrueTime tool. The subtask scheduling method is detailed in Sect. 3.

In the sequel, we specify the used FBS as well as the servo-motor and the pendulum processes to be controlled with a PID controller and finally specify the cost criterion of the QC.

2.2 Physical Processes

The first case study application concerns three second order processes. It consists of three similar servo-motors, each one described by the transfer function

$$G(s) = \frac{1000}{s(s+1)}\ . \tag{1}$$

We define by r the reference signal, y_i the measure of i^{th} process and u_i the control send to this process. The second case study consists of three inverted-pendulum which are a convolution of the inverted pendulums, carts, motors and the pulley chain mechanisms as specified by the transfer functions (Fig. 2). The Inverted pendulum is often considered as reference benchmark in control design problems. For our simulation, the pendulum starts from the center which corresponds to an angle of 0 rad. It will be constrained to an impulsion of 0.0873 rad (about 5°), applied on the cart two seconds after the beginning of the simulation.

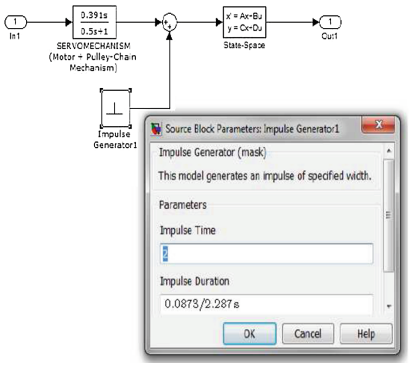


Fig. 2. The inverted pendulum, version on cart.

2.3 Feedback Scheduling and the EDF Protocol

The purpose of the used FBS, is to rescale periods and hence keeps the task system in a non-overloaded state. However, this advantage requires a particular executive (i.e., offering this complicated service to implement) and raises the problem of unpredictable latencies of tasks with lower priorities as stated in [1]. Job durations of the three controller tasks τ_1, τ_2 and τ_3 are generated according to a Weibull distribution as in [1]. This distribution is defined by three parameters: the localization parameter l which fixes the best case execution-time, the shape factor λ and the scale factor μ. Variation in task execution-times during the simulation is accompanied by task periods rescaling, in order to achieve an observed processor utilization associated to the used schedulability bound.

At the end of each job $\tau_i(\mathrm{k})$, the execution time $\hat{C}_i(\mathrm{k})$ is smoothed by a low pass filter. The FBS relies on this value, to calculate an estimate for the CPU utilization factor $\hat{U}(t) = \sum_{i=1}^{N} \hat{C}_i(k)/h_i(t)$. The EDF algorithm is also tested in this paper, because it is considered as an FBS that rescale periods when U is upper than 1. The average actual period of task τ_i in stationarity, $\bar{h}_i$, is given by $\bar{h}_i = h_i \mathrm{U}$ [2].

2.4 PID Controller

The PID controller defined by Eqs. (2–7) is used. This controller is developed in [22]. Given the fact that we rescale periods by FBS to ensure estimated schedulability, a_d and b_d parameters are recomputed according to formulas (5) and (6). Thus, a derivative term is computed using backward differences and a low pass filter (Eq. (4)) is used.

$$P(k) = K(\beta * r(k) - y(t_k)), \tag{2}$$

$$I(k) = I(k-1) + K * \frac{h}{T_i}(r(k) - y(t_k)), \tag{3}$$

$$D(k) = a_d * D(k-1) + b_d * (y(t_{k-1}) - y(t_k)), \tag{4}$$

$$a_d = \frac{T_d}{N * h + T_d}, \tag{5}$$

$$b_d = \frac{N * K * T_d}{N * h + T_d}, \tag{6}$$

$$u(k) = P(k) + I(k) + D(k). \tag{7}$$

PID parameters (K, T_i, T_d, N) are tuned in a way to obtain a system closed-loop bandwidth of $\omega_c = 20\ rd/s$ and a relative damping $\xi = 0.707$. This excludes the fact that the controller design and discretization may be a source of instability for the range of the sampling periods h_i. For such convergence the cost (8) has been specified to respect a threshold of 0.36. This outset for divergent costs is taken for a simulation time $\mathrm{T}_{sim} = 5\,\mathrm{ms}$.

$$J_{\mathrm{yr_i}} = \int_0^{\mathrm{T_{sim}}} |r - y_i| dt\ . \tag{8}$$

3 Subtask Scheduling

To simulate the subtask scheduling, the task model presented in Subsect. 2.1 is used. With a fixed priority assignment scheduling protocol, we assign the highest priority to the Calculate-Output segment (time critical part) and the lowest priority to the Update-State segment (must respect the period as deadline). It is obvious that the improvement will concern τ_3, the task which has the lowest priority.

3.1 Schedulability

It is noted in [2] that the ideal case of subtask scheduling under FP scheduling suggests that all Calculate-Output tasks segments have higher priorities than all Update-State tasks segments. Unfortunately, such priority assignment may

render the tasks system unschedulable. In cases where this approach does not work, an iterative algorithm is used. Given a schedulable original tasks system, the iterative algorithm attempts to minimize the deadlines of the Calculate-Output segments while maintaining schedulability.

In our work, the used FBS does not care about job overruns and the basic FP implementation technique of [2] is used. Since the TrueTime tool supports dynamic changes of priorities, we simply insert the TrueTime instruction "Set-Priority" in the code when entering a new segment (i.e., subtask in this model), see Listing 1. Note that the priority changes may introduce additional context switches, which can degrade the performance in a real system.

Listing 1. Implementation of subtask scheduling under fixed priority scheduling.

```
1  t := CurrentTime;
2  SetPriority(P_CO);
3  LOOP
4  ReadInput;
5  Calculate−Output;
6  WriteOutput;
7  SetPriority(P_US);
8  Update−State;
9  t := t + h;
10 SetPriority(P_CO);
11 SleepUntil(t);
12 END;
13 }
```

It has been established in [2] that the input-output latency Lio is reduced to 42% and the used cost (an LQG function based on the control and the output signals), is reduced up to 26%. Nevertheless, it is also noted that even if the latency is fixed and known, delay compensation can only recover part of the performance loss. This fact is illustrated by an example where the control cost of an integrator is given by $J \approx 0.79h + L$, for details, see [2].

4 Predictive-Delay Control

To improve the QC, the P-DC method brings up a predicted response time latency Lresp_i of the concerned task τ_i to calculate the control signal u_i. This artifice helps bypassing several practical problems like schedulability, convergence and computation time from which suffer most of proposed solutions. The method relies on an estimate Lresp_i, the current and the previous measures to extrapolate the forthcoming measure y_i required in the PID control calculus (Fig. 3). Without the P-DC, the measure to be used in the PID will be obsolete.

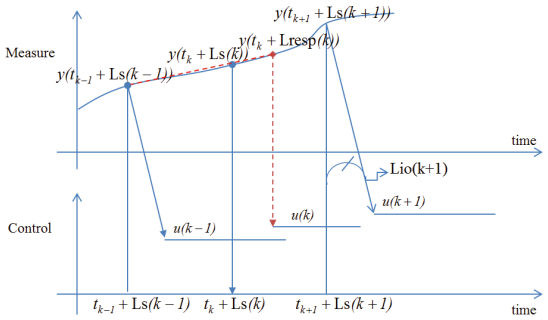


Fig. 3. Predictive measures based on Lio [1].

5 Overloaded System-Based Comparison Without FBS

In order to check the effect of some disadvantage of EDF and subtask techniques, under overload conditions, we experiment the tasks system defined in Table 1, where tasks τ_2 and τ_3 are perturbed (with lower priorities), and a utilization rate slightly overflowed U = 1.093 (Fig. 4g). Durations are given in ms. The shape factor μ is chosen high enough to ensure wide confident interval of the $C_i(k)$ values.

Table 1. The three servo-motors tasks system for an overloaded processor.

	h_i^{nom}	C_i	l	μ	λ
τ_1	6	4	3.7	0.0009	3
τ_2	13	4	3.7	0.0009	3
τ_3	14	4	3.7	0.0009	3

For most simulations, we noticed that when the FBS is turned off, the subtask algorithm outperforms in terms of improvement, but the QC values of the three solutions are close. The simplest solution for embedded implementation is the one for which only the prediction of the measurement, to improve the control (Fig. 4f), is used. The two other solutions, Subtask and EDF, induce two important problems in embedded design, such as the energy consumption due to the number of task context switching and the difficulty of managing the priorities in the queues of tasks. Another drawback is unplanned fairness between tasks which is contradicts specifications for systems where some tasks have to be prioritized over others. For instance, the task τ_2 is more penalized under EDF than under the two other solutions.

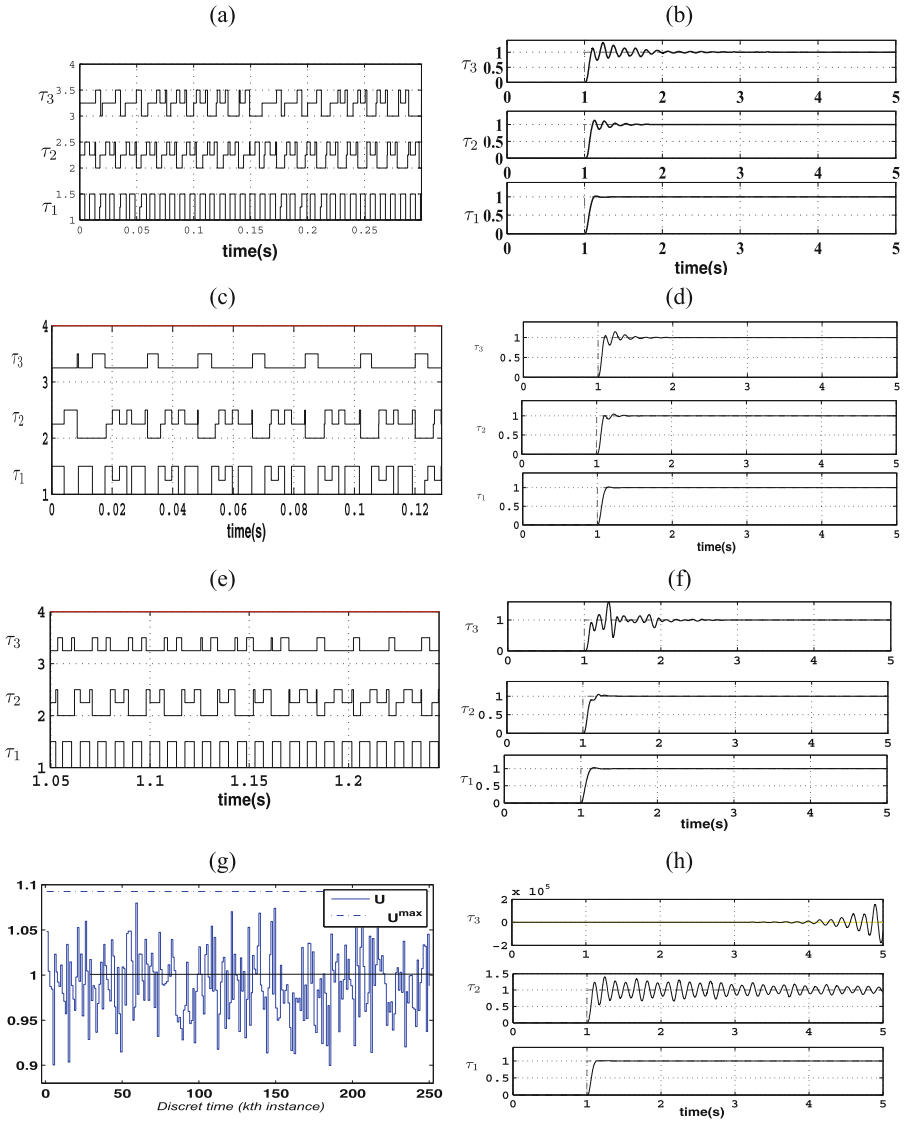


Fig. 4. Output y and tasks scheduling for an (h) overloaded processor. (a, b) Subtask, (c, d) EDF, (e, f) Predictive-Delay, (h) RM only.

It is important to note, that the use of the FBS only diverges the control, because small periods increase the latencies. Nevertheless, when the FBS is configured with a maximum utilization rate for EDF algorithm (U = 1), the best results are obtained for low priority tasks and equivalent for the others.

6 Comparison and Hybridization Under FBS and Scheduling Artefacts

6.1 Tasks Systems

The tasks systems used in the present section are described in Tables 2 and 3. The best case execution time l is turned down to 3.1 ms to have convergent and divergent costs by a large confident interval of $C_i(k)$.

Table 2. The three servo-motors tasks system for scheduling artifacts characterization.

	h_i^{nom}	C_i	l	μ	λ
τ_1	6	4	3.1	0.0009	3
τ_2	13	4	3.1	0.0009	3
τ_3	14	4	3.1	0.0009	3

The system defined in Table 3 is simulated with the same range of processor utilization U_i as in the three servo-motors example, where periods and execution times are both multiplied by a factor of 1.6. It is worth noting that the task sampling period never exceeds the divergence threshold of 27 ms for the servo-motor and 60 ms for the inverted pendulum. These thresholds are related to the PID setting described in the next subsection.

Table 3. The inverted pendulums tasks system characterization.

	h_i^{nom}	C_i	l	μ	λ
τ_1	9.6	7.5	5	0.0014	3
τ_2	20.8	7.5	5	0.0014	3
τ_3	22.4	7.5	5	0.0014	3

6.2 Impact of the Input-Output Latency on QC

For the tasks system presented in Table 2, the QC may diverge because of high input-output latency of lower priority tasks, due to preemption from tasks of higher priority level. Figure 5 confirms this behavior. The motor controlled by the task τ_3 diverges.

6.3 Case of Subtask Scheduling

Nevertheless, for overload situation, it can happen that τ_3 is blocked most of the time. Scheduling of this case is shown in Fig. 6. The output measure y_i for each task τ_i of the tasks system defined in Table 2 is shown in Fig. 7. Undesirable

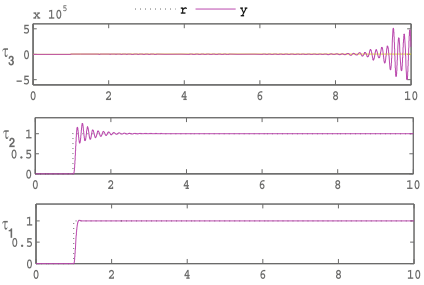


Fig. 5. The three servo-motors example with the subtask model and wide range of C_i(k).

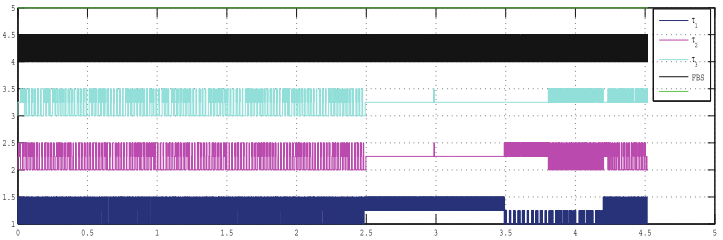


Fig. 6. Scheduling diagram of the three servo-motors example with the subtask scheduling method under overload situation.

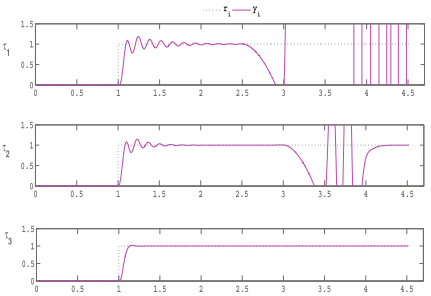


Fig. 7. Output measures of the three servo-motors example with the subtask scheduling method under overload situation.

breaks in the diagram testify the overload situation under subtask scheduling method. In this marginal case, tasks τ_2 and τ_3 are concerned within the interval times [2.5, 3.5] and [2.5 3.8], respectively. Figure 7 shows the divergence of tasks with lower priority τ_3 and then τ_2 as a consequence to the overload situation.

6.4 Case of P-DC Solution

With the observed C_i(k), within the overloaded case of the subtask simulation of Sect. 3 we obtain the P-DC result presented in Figs. 8 and 9.

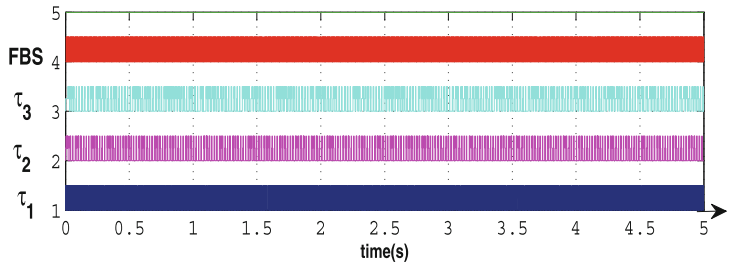


Fig. 8. Scheduling of three servo-motors under P-DC with an estimate Lresp.

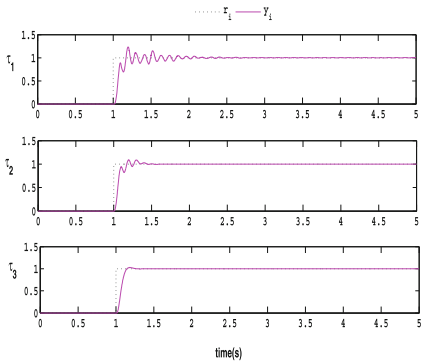


Fig. 9. The three servo-motors controls converge to the set point with a low cost for an overloaded system.

6.5 Hybridization

Table 4 sums up the comparison (about 2000 simulation samples) among six different implementation issues for the P-DC solution in the first column of each task. The first line is used for the ideal P-DC solution where the actual latency is used for the prediction of the output y. The last line is reserved for the hybrid solution proposed to enhance the QC of tasks τ_2 and τ_3. We observe that when the hybrid solution is not involved in the comparison tests, for mild to moderate deterioration as in the case of task τ_2, or for obvious deterioration like in the case of task τ_3, using estimated Lresp (line 4, 5) or its previous values (line 2 and 3), the P-DC solution may be helpful.

More precisely, in the case of the task τ_2 the performance is usually better when the Lio^{ub}(WCET) or Lio$^{ub}(\hat{C}_i(k))$ are used to predict the output. This

Table 4. Summary statement in comparison and hybridization between subtask scheduling and P-DC for the case of the three servo-motors.

		τ_2		τ_3	
		# wins [%] P-DC solely	# wins [%] hybridization involved	# wins [%] P-DC solely	# wins [%] hybridization involved
1	Actual Lio	0.2	0	70.7	36.3
2	Previous Lio	0	0	0	2.90
3	Previous Lresp	10.8	11.4	1.4	0.4
4	Lio^{ub}(WCET)	45.4	43.6	0	0.03
5	$\text{Lio}^{ub}(\hat{C}_i(\text{k}))$	43.5	44.6	0	0
6	$\hat{\text{Lio}}^{ub}(\hat{C}_i(\text{k}))$	0.1	0	15.2	4.16
7	Subtask only	0	0	12.7	0.13
8	Subtask & P-DC	0.8	0.4	0.8	56.06

remark does not necessarily mean that the Lio^{ub}(WCET) is longer, however, it may be more adequate than the other tested response times for the prediction of y_2. Meanwhile, this may mean that the task response-time is usable for the prediction, particularly in the case of moderate deteriorations.

In the case of τ_3, the subtask scheduling combined with P-DC leads to a solution that outperforms those obtained using P-DC or the subtask scheduling solely. Also, the smoothed and the previous Lio may be appropriate for the y_3 prediction.

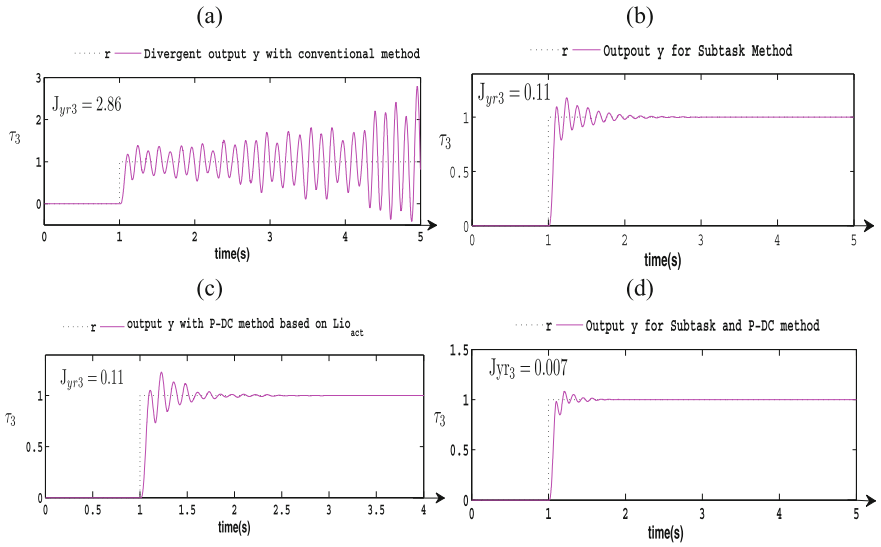


Fig. 10. Output and costs for a divergent QC (a) and for improved QC (b, c, d), case of the three servo-motors.

The obtained performances of 20000 simulations are carried out in the graphical simulation. It is observed that the improvement amounts of divergent controls (e.g., Fig. 10a), based either on the previous or on an estimated Lio which are computed on the basis of the previous and an estimated Lresp, respectively is sensibly the same. Figure 10c shows the QC improvement when using actual Lio in case of task τ_3. This result is not far from the improvement based on the subtask solution shown in Fig. 10b.

Implementation of solutions based on the previous Lresp or Lio needs system calls to save the response time and eventually the sampling latency for each job termination. However, solutions with the response-time calculated on the basis of upper bounds may show significant improvements.

To verify these results, we plot the Lio impact on the QC of the 20000 samples for each technique. Figure 11a shows the improvement of the QC when the previous value of Lio is used as an estimate. The result in Fig. 11b is based on actual Lio and is similar to the one obtained when the previous Lio is used.

The smoothed Lio in Fig. 11c, can be considered as the easiest prediction if we use a simple filter; the same as the one used to smooth the execution-time values. Similarly in Fig. 11d, the result gives the best cost where $J_{ry\,3}$ is all time

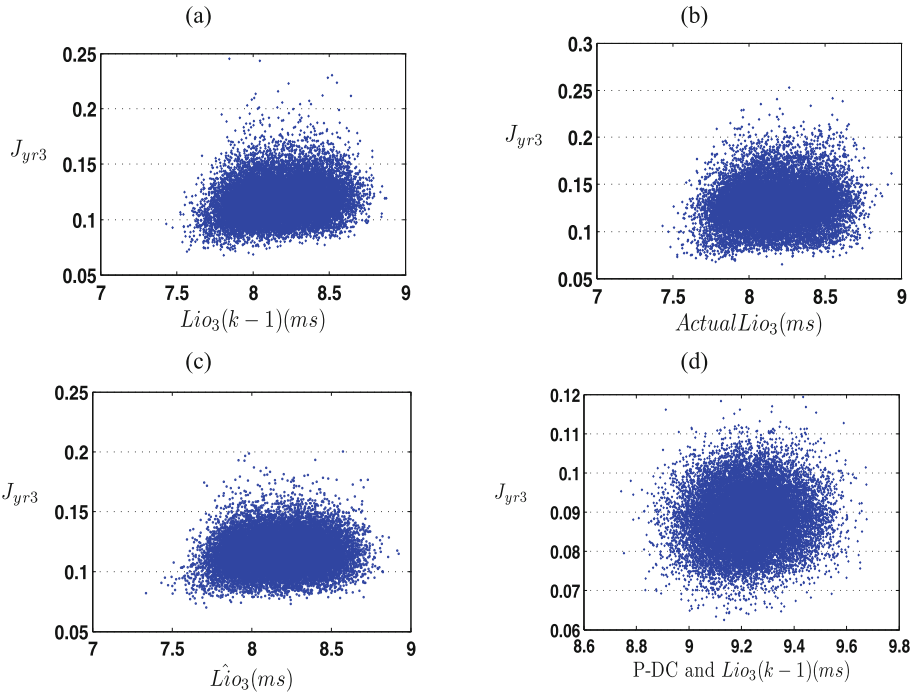


Fig. 11. Improved QC and performances comparison between proposed solutions, case of the three servo-motors.

under $0.12 < 0.15$. In all the tested cases, it is noticed that J_{ry_3} never exceeds the value of 0.36 which is considered as a threshold in our specification.

It is also important to recall that, due to the overload situation, it was very difficult to accomplish the 20000 simulations samples for subtask solution.

For the example of the inverted pendulum, Figs. 12a and b show the impact of the input-output latency on the QC for 20000 simulation samples of 5 s. The pendulum is considered as a benchmark with a more sensitive cost, where $J_{ry} < 0.09$ for convergent control situation. The Cost J_{ry_3} converges for all the samples, which confirms the result obtained for the first example of three servo-motors.

It is also noticed that the P-DC method is more appropriate for impulse response systems like in the pendulum case.

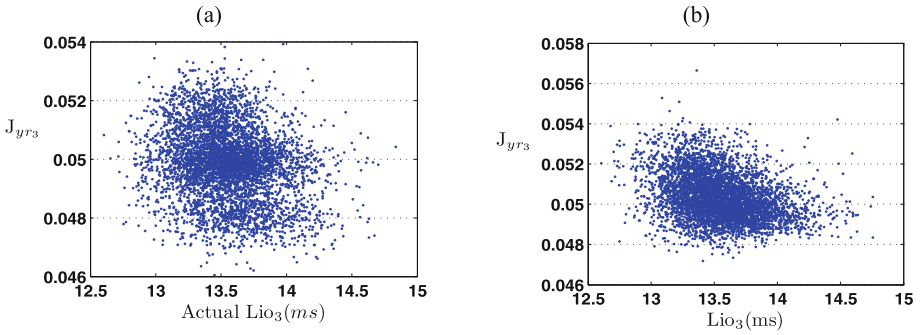


Fig. 12. Improved QC and performance comparison, case of the three inverted pendulum.

7 Conclusions

In this paper, predictive control simulation results are compared with those of naive control techniques. They show that predictive-delay method is able to improve performance of second and third order systems for a wide range of execution time in the presence of scheduling artefacts, while within the conventional executives (using fixed priority algorithms) it maintains the quality of control above the specified values. We found out that the hybridization of P-DC and subtask techniques, under an FP protocol, is a promising path that helps improving significantly the quality of the control. Indeed, hybridization can suggest a better quality than a scheduling or a feedback scheduling based solely on the predictive-delay control. Hence, it can be deduced that the predictive-delay control would be used not only to make up for scheduling latency but also to recover the control signal in overload situations. To sum up concluding remarks; reducing the input-output latency, through a subtask scheduling technique, can help boosting the P-DC method. For further works, we can compare the P-DC technique with other methods like the control server [23].

References

1. Sahraoui, Z., Grolleau, E., Mehdi, D., Ahmed-Nacer, M., Abdenour, L.: Predictive-delay control based on real-time feedback scheduling. Simul. Model. Pract. Theory **66**, 16–35 (2016). https://doi.org/10.1016/j.simpat.2016.02.013
2. Cervin, A.: Integrated control and real-time scheduling. Ph.D. thesis, Lund University (2003)
3. Sename, O., Simon, D., Ben Gaïd, M.E.M.: A LPV approach to control and real-time scheduling codesign: application to a robot-arm control. In: Proceedings of the 47th IEEE Conference on Decision and Control (CDC), Cancun, Mexico, pp. 4891–4897 (2008)
4. Robert, D., Sename, O., Simon, D.: An H_∞ LPV design for sampling varying controllers experimentation with a T-inverted pendulum. IEEE Trans. Contr. Syst. Technol. **18**, 741–749 (2010)
5. Xu, Y., Årzén, K.E., Bini, E., Cervin, A.: Response time driven design of control systems. In: Proceedings of the 19th International Federation of Automatic Control (IFAC) World Congress, Cape Town, South Africa (2014)
6. Cervin, A., Eker, J.: Feedback scheduling of control tasks. In: 2000 Proceedings of the 39th IEEE Conference on Decision and Control, vol. 5, pp. 4871–4876 (2000)
7. Seto, D., Lehoczky, J.P., Sha, L., Shin, K.G.: On task schedulability in real-time control systems. In: Proceedings of the 17th IEEE Real-Time Systems Symposium, Washington, D.C., USA, pp. 13–21 (1996)
8. Ryu, M., Hong, S., Saksena, M.: Streamlining real-time controller design: from performance specifications to end-to-end timing constraints. In: Proceedings of the Third IEEE Real-Time Technology and Applications Symposium (RTAS), Montreal, Canada, pp. 91–99 (1997)
9. Robert, D., Sename, O., Simon, D.: Sampling period dependent RST controller used in control/scheduling co-design. In: Proceedings of the 16th International Federation of Automatic Control (IFAC) World Conference, Czech Republic (2005)
10. Bini, E., Cervin, A.: Delay-aware period assignment in control systems. In: Proceedings of the Real-Time Systems Symposium (RTSS), Barcelona, Spain, pp. 291–300. IEEE (2008)
11. Yepez, J., Fuertes, J., Martí, P.: The large error first (LEF) scheduling policy for real-time control systems. In: Proceedings of the Real-Time Systems Symposium WIP, Cancun, Mexico, pp. 63–66 (2003)
12. Xia, F., Dai, X., Sun, Y., Shou, J.: Control oriented direct feedback scheduling. Int. J. Inf. Technol. **12**, 21–32 (2006)
13. Henriksson, D., Cervin, A., Åkesson, J., Årzén, K.E.: On dynamic real-time scheduling of model predictive controllers. In: Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, NV, vol. 2, pp. 1325–1330 (2002)
14. Henriksson, D., Åkesson, J.: Flexible Implementation of Model Predictive Control Using Sub-optimal Solutions. Institutionen för reglerteknik, Lunds tekniska högskola. Lund University (2004)
15. Cervin, A., Henriksson, D., Lincoln, B., Eker, J., Årzén, K.E.: How does control timing affect performance? Analysis and simulation of timing using Jitterbug and TrueTime. IEEE Contr. Syst. Mag. **23**, 16–30 (2003)
16. Gerber, R., Hong, S.: Semantics-based compiler transformations for enhanced schedulability. Citeseer (1993)
17. Gerber, R., Hong, S.: Slicing real-time programs for enhanced schedulability. ACM Trans. Program. Lang. Syst. (TOPLAS) **19**, 525–555 (1997)

18. Crespo, A., Ripoll, I., Albertos, P.: Reducing delays in RT control: the control action interval. In: Proceedings of the 14th IFAC World Congress, pp. 257–262 (1999)
19. Albertos, P., Crespo, A.: Real-time control of non-uniformly sampled systems. Contr. Eng. Pract. **7**, 445–458 (1999)
20. Balbastre, P., Ripoll, I., Crespo, A.: Control tasks delay reduction under static and dynamic scheduling policies. In: 2000 Proceedings of Seventh International Conference on Real-Time Computing Systems and Applications, pp. 522–526. IEEE (2000)
21. Åström, K.J., Wittenmark, B.: Computer-Controlled Systems, 3rd edn. Prentice-Hall Inc., Upper Saddle River (1997)
22. Åström, K.J., Hägglund, T.: PID Controllers: Theory, Design, and Tuning, 2nd edn. Instrument Society of America, Research Triangle Park (1995)
23. Aminifar, A., Bini, E., Eles, P., Peng, Z.: Designing bandwidth-efficient stabilizing control servers. In: Proceedings of 34th IEEE Real-Time Systems Symposium (RTSS), Vancouver, British Columbia, pp. 298–307. IEEE (2013)

Agent-Based Modelling and Simulation Framework for Health Care

Karam Mustapha(✉), Quentin Gilli, Jean-Marc Frayret, and Nadia Lahrichi

Mathematical and Industrial Engineering Department, Polytechnic University of Montreal, 2500, chemin de Polytechnique, Montreal H3T 1J4, Canada
{karam.mustapha, quentin.gilli, jean-marc.frayret, nadia.lahrichi}@polymtl.ca

Abstract. Colorectal cancer is a diagnosis of particular concern for older Canadians. Treatment of colorectal cancer requires a complex decision-making process of treatment. These treatments may involve surgery and either pre- or post-operative radiation or chemotherapy, which can have a great impact on the quality of life of patients due to the rigorous requirements of treatment and the inconvenient side effects. The conceptual and architectural modelling is challenging due to the diverse and complex dimensions. In this chapter we have proposed a modelling approach based on an additional structure to simplify the design of simulations. The modelling approach considers the complexity of the modelling process, where in the various models are developed. We developed a computer simulation environment of patient care trajectories using the agent in order to evaluate new approaches to increase hospital productivity and adapt hospital clinical practice conditions for the elderly and patients with multiple chronic diseases. So, we have developed a multi-agent framework to simulate the activities and roles in a Health Care (HC) system. This framework can be used to assist the collaborative scheduling of complex tasks that involve multiple personals and resources. In addition, it can be used to study the efficiency of the HC system and the influence of different policies.

Keywords: Health care · Agent-based simulation · Colorectal cancer

1 Introduction

Almost 88% of the Canadian population over the age of fifty[1] (41% women and 46% men) will develop some form of cancer during their lifetime. Lung, breast, colon, rectal and prostate cancers represent more than half of all new cancer cases (52%). Colorectal cancer is the third most common cancer among men and women and are considered the second leading cause of cancer death among men and the third among women.

Health Care (HC) is a rich domain for multifaceted simulation studies. The conceptual and architectural modelling is challenging due to the diverse and complex

[1] http://www.cancer.ca/~/media/cancer.ca/CW/cancer%20information/cancer%20101/Canadian%20cancer%20statistics/canadian-cancer-statistics-2013-FR.pdf.

M.S. Obaidat et al. (eds.), *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing 676,
https://doi.org/10.1007/978-3-319-69832-8_11

dimensions. In this domain, simulation generally aims at experimenting and testing management policies or organizational designs in a controlled environment in order to understand their economical, human and environmental consequences. This chapter deals with the simulation of cancer patients' pathways. With the aging population and the intricacy of the medical system, the management of HC activities has become increasingly complex. Therefore, simulation is a relevant tool to model this complexity and improve its operations. In particular, agent-based modelling and simulation significantly extend the capabilities of simulation approaches such as discrete-event simulation as discussed in the next section.

Providing high-quality care is a priority among health professionals. However, resources are limited and their utilization must be optimized in order to meet high quality standards and patients' unique profiles. Therefore, the challenge faced by HC providers and managers is to design organizational and medical processes that deliver the right treatment, to the right patient, at the right time using the right resources. Factors, such as socio-demographic and environmental characteristics, as well as the characteristics of the organizational and decision-making systems, can be used to simulate patient care trajectories, from their diagnosis to the end of the treatment.

In this chapter, we propose to study the efficiency of organization decisions which aims at: (i) describing the HC organization; (ii) modelling and simulating the behaviours and decisions of its actors and (iii) implementing these decisions and observe their local and global effect on the HC, and (iv) supporting each step with specific conceptual and software support.

This chapter presents different objectives, the first objective presented requires the agent-based modelling and simulation of complex behaviours, decision-making processes and interactions between hospital staff and patients. The most appropriate technology to simulate these complex mechanisms is Agent-Based modelling and Simulation (ABS). The second objective is therefore to create and validate the patient agent model, which includes a physiological model of how the cancer evolves in time in response to specific treatments. Also, to simulate a large number of patients treated simultaneously with the same resources of the hospital; this step of the project is only concerned with the general behaviour of the patient agent, and how well it can be configured in order to simulate colon and colorectal cancer patients with different attributes. As for third objective, healthcare decision makers need reliable tools to support them in decision making for adapting policies to help cut costs or reduce waiting time, and to provide visualization which allows them to rehearse innovative ideas before they are implemented.

Contributing to aforementioned objectives, we aim to developing a computer simulation environment of patient care trajectories using the agent in order to evaluate new approaches to increase hospital productivity and adapt hospital clinical practice conditions for the elderly and patients with multiple chronic diseases. Ultimately, the simulation model will include: the physical health of the patient; the cognitive state of the patient; the psychosocial state of the patient; the hospital resources, staff and physicians. For that, we have developed a multi-agent architecture to simulate the activities and roles in a HC system. This architecture can be used to assist the collaborative scheduling of complex tasks that involve multiple personals and resources.

In addition, it can be used to study the efficiency of the HC system and the influence of different policies.

So, this chapter describes the general scope of this simulation project and presents an up to date ABS. Next, the general conceptual model of the simulation is described and finally simulation results are presented.

2 Literature Review

Many research projects are based on the agent paradigm to model and/or simulate complex systems. Indeed, this paradigm provides a tailored approach to model complex systems by explicitly addressing the study of the interactions and behaviours of their components. The design of HC agent-oriented models is a difficult task that requires the use of specific knowledge and skills. This section defines ABS and introduces a detailed analysis of ABS applications in the medical domain. Finally, this section also presents different ABS development framework.

2.1 Agent-Based Simulation

ABS is an abstract representation of reality that involves the elaboration of a descriptive model, which reproduces the behaviour of the system by modelling its components, including their decision-making capabilities and interactions patterns, as agents. An agent can be defined as an entity, theoretical, virtual or physical, capable of acting on itself and on the environment in which it evolves, and capable of communicating with other agents [22].

Research in ABS is prolific. It is known under different labels, including multi-agent simulation, individual-based models and agent-based models. These tools are part of a more generic technology known as multi-agent systems; this domain of applications is much larger than simulation. In the literature, the concept of agent is generally defined as [22] "…a computer system situated in an environment, which is a way autonomous and flexible to achieve the objectives for which it was designed."

In practice, the multi-agent paradigm is used at two levels: for modelling and for simulating. At the first level, it is required to create multi-agent models that (1) reproduce the naturally distributed structure of the studied systems, or (2) propose a representation of complex problems. Such models can be used for developing reactive, deliberative or hybrid agent models. The second level involves the simulation (i.e. experimentation with these models). Such a simulation may or may not be based on distributed software architecture. In other words, the operational simulation model is not necessarily multi-agent. It may be object-oriented or translated into other simulation languages (e.g. DEVS).

2.2 Agent-Based Simulation in the Health Care Domain

HC operation management is a domain that is well suited to ABS because it involves many people interacting with their own decision-processes. With agent-based modelling, it is possible to explicitly model these individuals and their interactions.

However, although ABS is growing in the medical domain, applications to the real world are still rare [5, 28].

In the medical domain, [27] identifies 200 papers, in which simulation is used. More than 70% of these applications used Monte Carlo simulation, while 20% used Discrete-Event Simulation, less than 9% used System Dynamics, and finally only 1% used ABS.

For instance, [34] uses ABS to reorganize hospital emergency departments. Recently, several simulation techniques have been used in conjunction to capture different dimensions. [23] Use DES and ABS to model a healthcare system, in which patients choose their hospital based on a linear additive service function of three factors (i.e., hospital reputation, travel distance, waiting time). Finally, [7] proposes one of the first systematic studies aiming at comparing SD and ABS based on a simple mathematical model of interactions between a tumour and immune cells. The authors concluded that both modelling paradigms are not always equivalent.

In most organizational simulations in the medical field, agents, whether patients, doctors or nurses are of reactive type and their behaviour is very specific to the purpose of the simulation. In [21], the author discussed the introduction of a multi-agent system into the medical field, which helps the management take decisions and actions, and also ensures the communication and coordination by reducing the errors of diagnosis and treatment, and by improving time required for the medical resources, and other medical departments. However, [20, 25] use simulation in order to analyse the performance of an emergency department in different configurations. In these studies, agents are used to model resources that move through the hospital with predefined process time. In [19], modelling deals mainly with the different types of treatment associated with their time and resource requirements, which then become predefined in the simulation. Only patient's arrival time and resource availabilities change dynamically. In these models, the agents travelling times within the hospital is predefined. However, it can also be dynamically computed in the simulation as in [39], which models the evacuation of a hospital undergoing a fire, or in [24] that use simulation to study different transport configurations for clean and dirty equipment in the hospital.

Also, some authors proposed the concept of an online medical service system for internet users using a multi agent system, the user can get access to the details of the closest and best health care system such as hospital, medical clinic, etc. [10]. However, [14] used the medical sensor modules with combinations of wireless telecommunication technology based in the multi-agent system. The papers [12, 16], proposed a hybrid system with human and artificial agent members. [12] Proposed an operational algorithm to describe the operations of a hybrid multi agent system based intelligent medical diagnosis system called Clinical Diagnosis System (CDS) [13]. Also, [26] presented hybrid architecture of a multi-agent consultation system for obesity oriented health problems.

Some authors propose a multi-agent oriented learning environment aimed at learning using a positive approach to perform diagnostic reasoning and modelling of a domain [30]. In [3], the authors proposed the model of practical data mining diagnostic which intends to support real medical diagnosis by two emerging technologies - data mining [40] and multi-agent system [6, 22]. In the next section we present the patient agent models.

3 Patient Agent Models

In the literature review, we have presented various research based on the definition of methodologies to guide the designers in the development of multi-agent models in general. However, they present a number of weaknesses related to modelling HC, and their simulation, for example, at present, there is no generally accepted health care ontology for generating and analysis of medical or health care information. This makes it difficult to communicate between several systems developed in different areas. Also, other limitations related to the framework can be synthesized by the following: (i) the absence of an approach which ensures the passage of the conceptual level to the implementation level; (ii) the transition from design to implementation is costly in time and development efforts; (iii) consideration of the organization; (iv) multi-modelling and (v) time management.

In this study we proposed a modelling approach based on an additional structure to consider the complexity of the modelling process, where in the various models is developed. The real system is first represented by the HC domain and then the overall modelling approach is based on an incremental approach in which different models are developed. The expert's fields of intervention are specified including different models: conceptual modelling and simulation oriented agents.

In the process of modelling and simulation patient trajectory, the model distinguishes three main steps: the conceptual modelling, the conceptual agent modelling, and operation modelling (Simulation Oriented Agents - SOA).

The patient is the central actor of the HC system or real system. It interacts with many resources, including physicians, nurses and equipment. Its dynamic condition is the main driver of resource utilization, and its reaction to treatment defines the system quality level. In order to design such an agent, different models are proposed to describe its place in the overall system, and its complex behaviour.

Conceptual modelling is based on several models specifying the nature of the agents and the architecture of multi-agent system. In the following, it is for the programmer to operationalize the conceptual model agent. Each agent identified at the conceptual level is specified and implemented according to the constraints related to the development environment. It is always for the programmer, to take into account the technical constraints ignored at the simulation. Thus, the multi-agent system will be deployed in a software environment enabling its execution to conduct simulation experiments.

The conceptual modelling and conceptual agent modelling are described next. With regard to operating step, it will be addressed in the next section on the architecture simulation support.

3.1 Conceptual Model

The general conceptual model proposed in this study defines the main interactions between the patient and its environment (Fig. 1). It is composed of four dimensions and includes different aspects of the patient, its environment, and the HC system. These dimensions are related to physiology of the patient, the psychosocial state and support of the patient, the decision processes and the resources used to treat the patient. The links between the different aspects identified within these four dimensions represent

their mutual dependencies. The central part represents the patient agent. The other parts represent the hospital staff involved in the treatment selection, as well as patient support (e.g., family members, nurses) (for more details see the Ref. [9]).

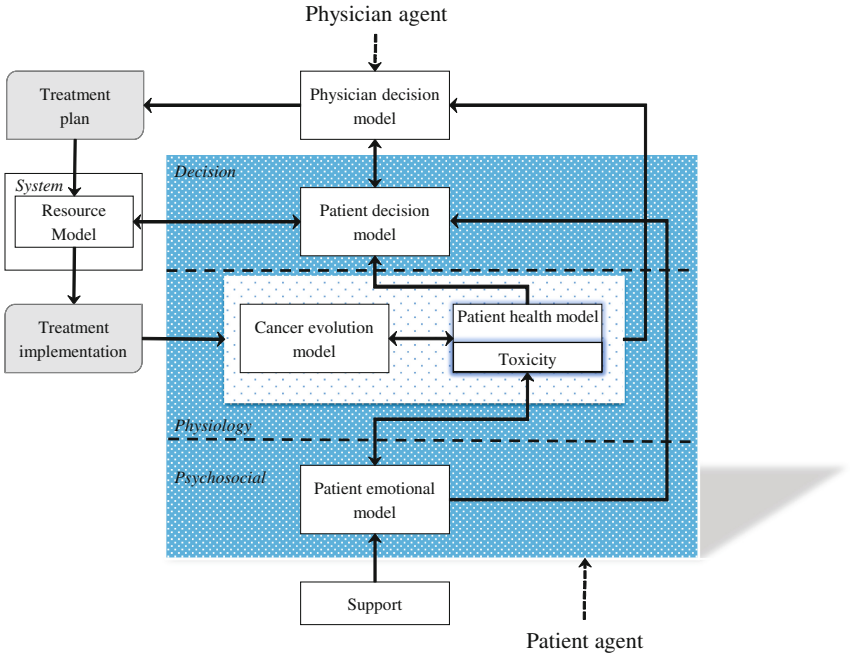


Fig. 1. General conceptual model.

Psychosocial Dimension. The psychological dimension includes an emotional model of the patient agent and its social influences, especially in the form of support from family members and nurses. This model describes a response to specific situations and will eventually contribute to measuring the patient quality of life during treatment.

Physiological Dimension. This dimension includes both the patient's health model (its general physical and health condition) and its cancer evolution model. Both are affected by treatment in different manners, while influencing each other. In practice, this dimension includes on the one hand, the absolute physiological state of the patient and cancer, and, on the other hand, the perception of this state obtained from observations (e.g., analysis, scans, and biopsies). While the first information is not necessarily known, the second can be out-dated, and more or less accurate. The variable describing the cancer evolution model in particular is described in the next. Finally, in this model, the patient health model is influenced by his or her emotional model.

Decision Dimension. This dimension includes both the patient's and the physician's decision models. It represents the main actors' decision-making processes and preferences that contribute to treatment selection and treatment implementation. It is the

part of the conceptual model that directly contributes to the decision and implementation of patient care trajectories. Here, the patient decision model is influenced by its health and emotional models, while the physician decision model is influenced by the patient cancer and health models. The patient decision model also contributes to plan each individual treatment according to the system resource availabilities.

System Dimension. The system dimension represents the virtual hospital resources and processes. When a physician requests a type of treatment, it must be planned according to the hospital priority, the workload of the resources required for this kind of treatment, as well as the preferences of the patient. The different sub-models of these dimensions' influence each other in order to emulate the general relationships between the patient, his/her cancer, the medical staff, and the patient's support. The relationship between the patient and the hospital processes and resources are addressed through the dynamic specification of the treatment program into the care trajectories, which defines how the patient interacts with the different resources for his/her treatment and tests/scans. The next section focuses on the conceptual agent model.

3.2 Cancer Evolution Model

Modeling the evolution of cancer is an important step for the simulation of care trajectories. In order to do this, the cancer will be modeled into two parts, the main tumor and metastases. Metastases are meant as a general term referring to every cancerous cell found in the patient's body that are not part of the main tumor. This may be an isolated cell traveling in the patient's body or a small tumor hooked somewhere else than the main tumor. The main tumor size and the number of metastases are two important information as they influence the decision about the treatment [1]. Both will be simulated from their appearance (size 1 cell for the tumor and no metastasis) to the end of the treatment. It is useful to model the evolution of the cancer before the diagnosis so that out of treatment evolution parameters can be validated and the distribution of metastasis density can be known. The evolution of the tumor model that will be described later has four parts: a free evolution and the three evolutions under each of the three treatments, which are radiation therapy, surgery and chemotherapy developments.

For the metastases evolution model, there are only two parts: as for the tumor model, a free evolution and an evolution under chemotherapy. There is no special evolution under radiation therapy because it has no effect on metastasis other than to reduce the emission of cancerous cells by the main tumor.

- ***Tumor Free Growth***

There is a lot of mathematical models of tumor growth based essentially on population-based models [36]. The original population-based model was developed by Maltus at the end of the 18th century, using Eq. (1):

$$\underbrace{Variation}_{X_p'(} = \underbrace{Number\ of\ birth - Number\ of\ death}_{g(X_p(t))} \tag{1}$$

here Xp(t) is the tumor size over time given in numbers of cells. One of the most common formulas used for g(x) is an empirical law (see Eq. 2) described by Gompertz in 1825 [36], which describes the evolution of the main tumor from the appearance of the first cancerous cell to a larger tumor.

$$g(x) = a * x * \ln\left(\frac{b}{x}\right) \tag{2}$$

with a being the rate of tumor growth (it is related to doubling time (DT) of the tumor); b is a constant equals 10^{12} and represents a maximum diameter of 12.4 cm (this value is used in most studies on solid tumours). Other tumour growth models exist, such as logistic and exponential models. The Gyllenberg-Webb model divides the evolution of the tumor in different phases depending on its size in order to describe its evolution more precisely [11].

In the simulation, the Gompertzian formula for the tumor free evolution was used. In order to determine a, we used [2], which characterizes the tumor growth of 27 patients suffering from colorectal cancer. Using this empirical study, we computed a Weibull distribution law of the doubling time, from which we randomly generated a doubling time DT. Assuming that this doubling time is a constant over the tumor growth, this allows us to calculate the time it takes for the tumor to be a given percentage P of the maximum size b using Eq. (3).

$$TumourSize(t) = e^{\frac{\ln(2)}{DT} * t} \tag{3}$$

Once the T is known, a is calculated using the Gompertz curve function, therefore, the link between the doubling time and a can be calculated using Eq. (4).

$$a = -\frac{\ln\left(-\frac{\ln(P)}{\ln(b)}\right) * \frac{\ln(2)}{DT}}{\ln(P * b)} \tag{4}$$

- ***Tumor Growth After Radiation Therapy***

First, only external radiation therapy is modeled. Its impact on the size of the tumor is calculated one session at a time. Consequently, the remaining number of cells is the number of cells before treatment multiplied by the percentage of surviving cells S represented by Eq. (5) from [38].

$$S = e^{-A\left(\alpha * d + \beta * d^2\right) + B} \tag{5}$$

with α and β being constants for colon and colorectal cancer, respectively 0.339 and 0.067, as empirically estimated by [35]; d is the dose used during the session; and A and B are two parameters associated with the patient, corresponding to the effect of a variety of factors. They follow a normal distribution determined using [35]. This model is based on two assumptions. First, each cell that cannot further divide itself after the radiation therapy session, is considered dead. Second, the tumor keeps growing freely between sessions.

- ***Tumor Growth During Chemotherapy***

The action of chemotherapy is determined using a model developed and tested with two types of chemotherapy drug (i.e., Fluorouracil and Capecitabine) on colon and colorectal cancer [4]. Based on this study, the function g(x) in Eq. (1) is described by Eqs. (6) and (7):

$$g(x) = (a_c - E(t)) * x \tag{6}$$

$$E(t) = E0 * \sum\nolimits_i Concentration(t, T_i) \tag{7}$$

With a_c being the exponential growth factor of the tumor. It is determined according to the parameters of the Gompertzian growth and the tumor size at the beginning of chemotherapy. Concentration (t, Ti) represents the function of drug concentration injected at time T during session i, in the patient's body over time. E0 is the effect of the drug on the decrease of the tumor [36]. E0 depends on the patient and on the type of treatment. We model three types of drug administration: Oral, injection with syringe and long injection like Portacaths [32] and Piccline [33]. The function of concentration of drug in the patient's body over time is different for these three types of administration (see Eqs. 8, 9 and 10), based on [4, 36].

For Injection with Syringe and Oral Administration

$$Concentration(t, T_i) = Dose * \left(\frac{1}{2} + \frac{1}{2} * tanh(k(t - T_i))\right) * e^{(Absorption*(T_i - t))} \tag{8}$$

with Absorption being the speed of drug elimination from the patient's body; k is the speed of drug assimilation; and Dose is the dose injected during the session. The only different between injection with syringe and oral administration is k, which is bigger for injection.

For Long Injection

Concerning long injection, the only new parameter is duration, which is the length of time of the injection, as shown in Eq. (9).

$$\begin{aligned} Concentration(t, T_i) = Dose * \left(\frac{1}{2} + \frac{1}{2} * tanh(k(t - T_i))\right) \\ * \; e^{\left(\frac{1}{2} + \frac{1}{2} * tanh(10(t - T_i - duration))\right) * (Absorption * (T_i - t + duration))} \end{aligned} \tag{9}$$

Finally, the function of the tumor's size during chemotherapy is:

$$X_{pc}(S_{T_c}, t) = S_{T_c} e^{\int_{T_c}^{t} (a_c - E(s))ds} \quad with\, t > T_c \tag{10}$$

with S_{T_c} is the tumour's size before the beginning of chemotherapy. This value is also used in the metastatic evolution model.

- ***Tumor Growth After Surgery***

The effect of surgery on the size of the tumor is simpler than the other two treatments described above. Indeed, depending on the cancer (colon or rectum) and the type of surgery, the effect of the surgery can be described as a probability of having cancerous cells from the main tumor remaining in the body. The next section presents an illustrative example of a cancer patient treated with two treatments.

- ***Metastases Growth***

For the development of metastases, we use a model developed by Iwata ([17]). In this model, the growth of main tumor and the metastases are described by a set of mathematical equations. The tumor growth is modeled by $X_p(t)$, which can either be the Gompertzian function (2) or the exponential function (3). Next metastases growth, produced by the main tumor and other metastases, is described by Eq. (11).

$$\beta(x) = m * x^{\alpha 2} \tag{11}$$

with m being the coefficient of colonization, and α2 being the fractal dimension of blood vessels infiltrating the tumor.

Considering that all tumors evolve similarly is not entirely correct. Indeed, although they all originate from the main tumor (i.e., their nature is similar), their spread and evolution depend on their location. However, accurately modeled movement of each tumor cell in the body is impossible. The Iwata model and its assumptions are considered valid and used in the majority of evolution models of metastases. Iwata's model is defined by the system of Eq. (12):

$$\begin{cases} \frac{\partial \rho(x,t)}{\partial t} + \frac{\partial g(x) * \rho(x,t)}{\partial x} = 0, \\ \rho(x,0) = 0, \\ g(1) * \rho(1,t) = \int_1^{\infty} \beta(x) * \rho(x,t) dx + \beta\big(X_p(t)\big). \end{cases} \tag{12}$$

with $\rho(x,t)$ being the density of metastases in the patient's body (i.e., the number of tumors containing x cells at time t), and $g(x)$ being the function defined above. Both parameters m and α2 are specific to each patient and have a normal distribution, which are determined thanks to [1, 17].

The value of interest for the decision-making is the Metastatic Index (MIn). It is defined by Eq. (13) in [1]. It represents the total number of metastatic tumors of size between n and $X_p(t)$ in the patient's body at time T.

$$MI_n(T) = \int_n^{X_p(T)} \rho(x,T) dx \tag{13}$$

The resolution of the Iwata model is more complex than that of the primary tumor. Furthermore, there is general solution of this model with a function $g(x)$ with chemotherapy. Therefore, in order to keep calculation time reasonable within the simulation environment, the Iwata model is only used to describe the evolution of metastases without treatment using function $g(x)$ defined in Eq. (2).

- ***Metastases Growth During Chemotherapy:***

Granted there is no general solution to the Iwata model with chemotherapy, in order to determine the effects of chemotherapy on metastases, we first made three assumptions:

Cancer cell dispersion in the body (i.e., $\beta(x)$) is neglected. Because cancer cell progressing through the patient's body is directly in contact with the drug, we assume it is automatically destroyed.

All metastatic tumors evolve along the same decay law as the primary tumor under chemotherapy. In this study, we use Eq. (6):

The number of tumors given by $\rho(x, T_c)$, as defined in (12) at the end of the free evolution of metastases, remains unchanged during the chemotherapy treatment. Only the tumour's size is affected.

Based on these hypothesis, the new distribution of metastases during chemotherapy $(t > T_c)$ can be calculated based on $\rho(x, T_c)$ as defined at the end of the free evolution of metastases, using Eq. (14):

$$\rho(X_1, t) = \rho(X_2, T_c) \quad with \quad X_1 = X_{pc}(X_2, t) \tag{14}$$

We define a new function X_{pc}^{-1} as.

$$X_{pc}^{-1}(X_1, t) = X_2 = \frac{X_1}{e^{\int_{T_c}^{t}(a_c - E(s))ds}} \tag{15}$$

Therefore, MI during chemotherapy can be calculated using Eq. (15):

$$MI_n(t) = \int_n^{X_{pc}\left(X_p(T_c),t\right)} \rho\left(X_{pc}^{-1}(x, t), T_c\right) dx \tag{16}$$

3.3 Conceptual Agent Model

The conceptual agent model must determine a number of properties of the previous conceptual model presented. Focusing mainly on aspects of design and analysis, the conceptual agent model integrates the major concepts of agent, role, service and relationship, defined as:

- The agent is an active entity of the environment;
- The role is played by the concept of an agent;
- A service is a function performed by an agent;
- A relationship is an interaction between entities.

The concepts behind the conceptual agent model are defined through a meta-model. This meta-model defines as precisely as possible all the concepts involved in a conceptual agent model and semantic relationships. The conceptual agent meta-model is formalized by the UML class diagram shown in Fig. 2.

In this conceptual model presented an agent plays different roles. The same role can be played by several agents. A role provides services, while a service may require a

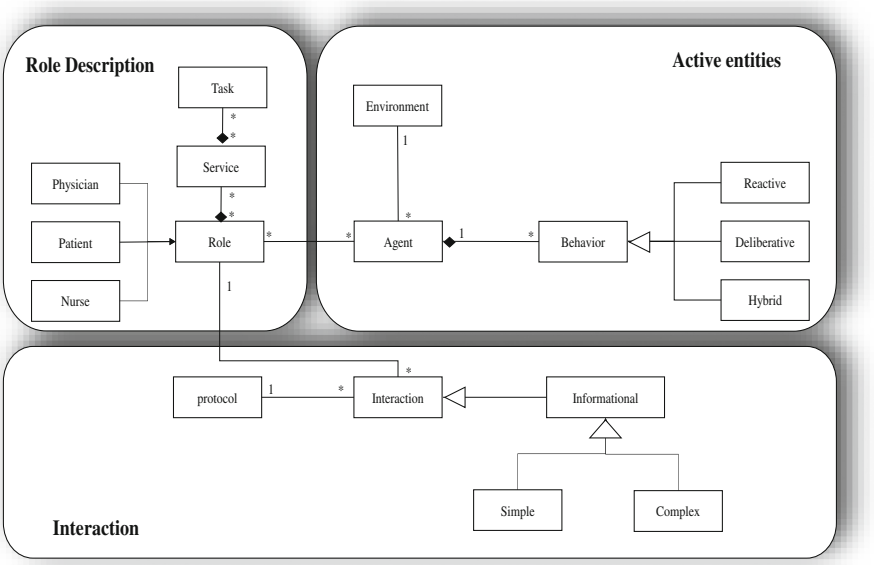


Fig. 2. Conceptual agent model.

task. Relationships can develop between roles. There are two sub-types of interactions, simple and complex interactions (informational). The simple relationship is an exchange of information to complete tasks, the distribution of tasks or the sharing of knowledge and the complex relationship for example assumes that agents must coordinate their actions in order to combine their skills to solve complex tasks. An interaction composed protocol. Finally, there are several types of agents: reactive (If the simple behavior is required, a type of stimulus-response behavior is sufficient), deliberative (If decision making and negotiation are needed, it will be the capacities of a deliberative agent to perceive its environment and the behavior of other agents), hybrid (Reactive behavior and deliberative behaviors are needed. For example, an agent "smart" capable of interacting with another agent when disruptive events occur). In the next section we present the Simulation of Care Pathways for Patients (SiCaPP).

4 Simulation Methodology (SiCaPP)

The objective of this section is to present the software solution restraint to accompany the process design and Simulation of Care Pathways for Patients (SiCaPP) for colon and rectal cancer treatment by integrating the functional and software requirements, and based on multi-agent modelling.

SiCaPP represents an implementation solution for the conceptual agent model and is characterized by:

- Specification, the agents' behaviour in appropriate languages to the granularity of agents, it is to describe how the agent should behave during the simulation without prejudging how they will actually be implemented (language programming, simulation language, environment, etc.)
- The specification of interactions between agents which results in dynamic simulation. These interactions will have different implementation issues that are involved as agents of a same environment.

The simulation environment aims both to facilitate the handling of models and supervise their implementation in order to exploit their results.

5 SiCaPP Architecture

SiCaPP architecture presents different services, these services include the following information: agents' management, time management, and inter-agent communication. The agents' management provides all the functions needed to manage the life cycle of agents addressing, functions such as launch and stop. It allows for example, adding, changing, or deleting the agents dynamically, it maintains a directory of these agents taking particular account of the simulator in which they operate. Secondly, the inter-agent communication presents different communication languages like ACL message and provides the communication between agents in the environment. It can also manage a directory like yellow pages integrating information on the capabilities and/or agent played roles of the simulation. Finally, the time management is rarely mentioned in multi-agent, of the fact that the distributed nature of the simulation is often more conceptual than software. Thus, time management is implicitly centralized on the reactive multi agent system and is not managed in the deliberative systems if not in relative terms.

In this architecture, we also define a different role that includes the following information:

- A set of actions that can be performed, i.e. a patient role is to approve action prescribed by physician.
- A set of protocols, which describe how this role should interact with other roles.
- A set of goals.

The SiCaPP system is organized into different packages (Fig. 3), packages include:

- User interface: It is a GUI-based logic which enables the user to generate, simulate patients and show simulation results graphically.
- Database: It is used to register generated patients and the simulation results.
- Patient population: It generates patient's population; this information includes aspects of a patient's personnel information and physical health such as treatment plan, medication and diagnosis.

- Patient simulation: Controlled by physicians who decide whether diagnostics are to be accepted, perform medical and surgical interventions, provide prescriptions, and perform chemotherapy and radiotherapy treatment in collaboration with nurse.
- Treatments protocols: It describes a method to be used during the treatment (e.g. drug, medical treatment) or a medical research study.
- Treatment plan: This package is used to choose a treatment trajectory plan for patients based on the epidemiological studies and real data.
- Equation methods: it contains different sets of mathematical equations used in our model, e.g. gompertz model which describes the evolution of the main tumour from the appearance of the first cancerous cell to a larger tumour, iwata [6, 17] model which is used to describe the evolution of metastases.
- Clinical profile: It is used to check the physical and psychological state of the patient, it is based on different notions e.g. depression, status performance, sleep disturbance and fatigue. We are using the epidemiological studies and Jewish Hospital real data.

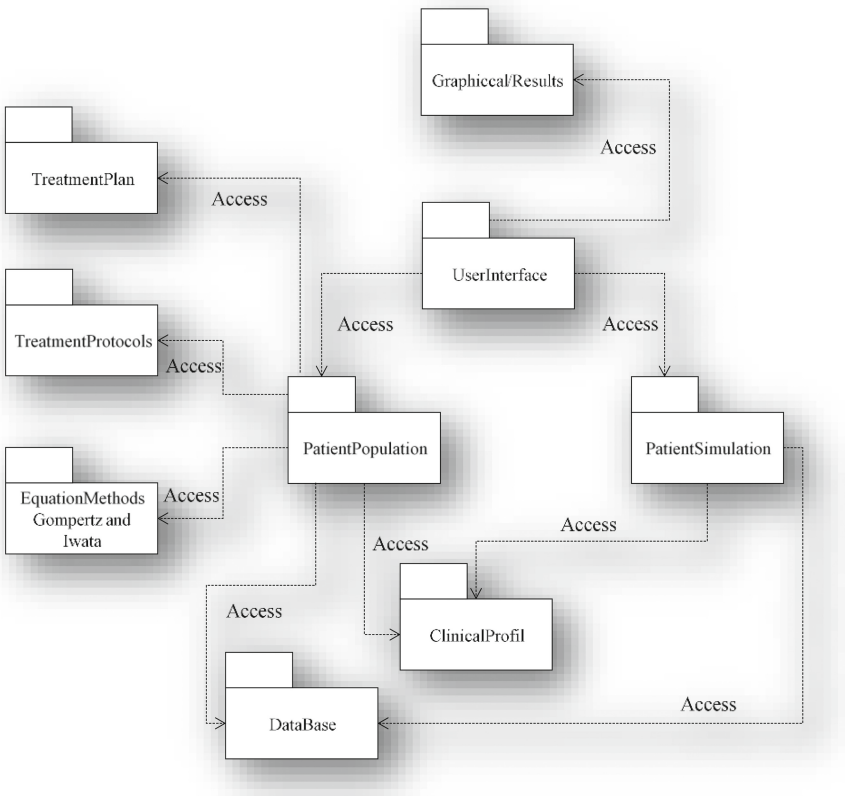


Fig. 3. SiCaPP architecture diagram.

6 SiCaPP Kernel

Medical information of a patient is one of the most sensitive types of information; this information includes aspects of patient's personnel information and physical health such as treatments, medicines and diagnosis. A patient may be treated by any number of physicians or nurses but they must all belong in the team which is responsible for this patient. A physician can treat any number of patients and maintain the medical history for each patient (see Fig. 4).

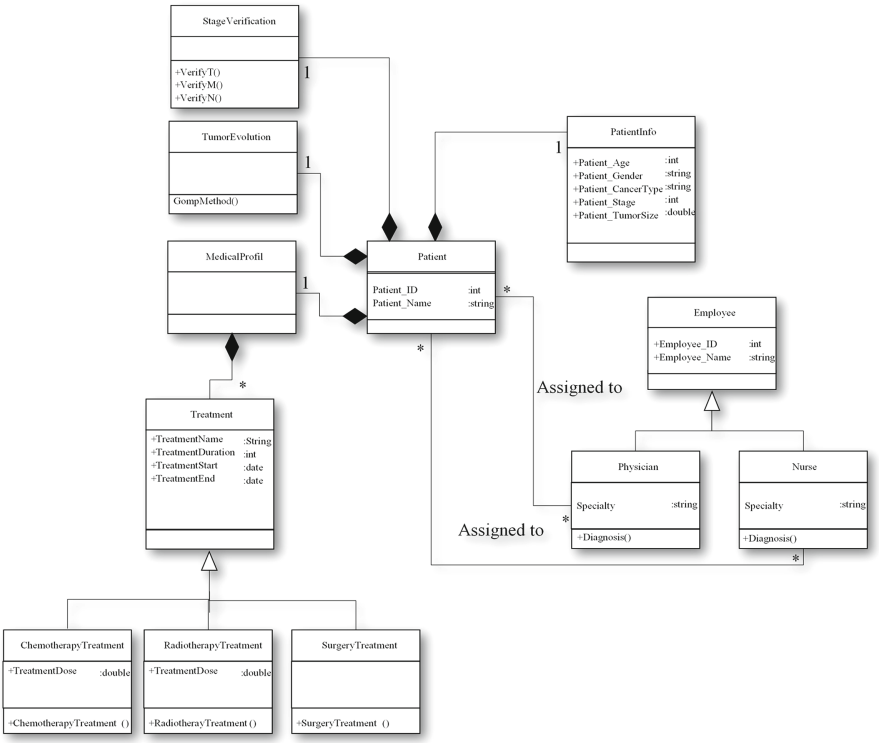


Fig. 4. Patient class diagram.

The patient is considered as a composed class to calculate tumour evolution using mathematical models. Tumour growth is based essentially on population-based models [36]. Also this class is used to verify the stage before and during the treatment (diagnosis step). Each patient has a medical profile; this profile contains a record of all treatments used within the medical group. If the patient has been treated in any facility within the same medical group, we will have an existing patient record and a medical history for the patient; this may need to be updated. A treatment instance is created for all patients admitted and updated throughout the patient's stay. The treatment will subsequently be added to the patients' medical record upon patient discharge.

6.1 Generate Population

Based on epidemiological studies and the real data, two different methods are used to generate the population of patient. Figures below show a state chart for the class to generate a virtual patient population. Firstly, this class generates the age and gender using the epidemiological studies (Fig. 5) alternatively we can extract this information from real data (Fig. 6). Secondly, based on the age and gender we choose the cancer type (two types are available: colon and rectal) and stage. Alternatively, we can also extract this information from real data. Finally, we use the gompertz model to determine the tumour size in mm and we can calculate the stage using the iwata model. If the stage obtained is different using both models we have to re-determine the stage again and repeat the same procedures. In case stage results are matching, population generated is registered in the data base.

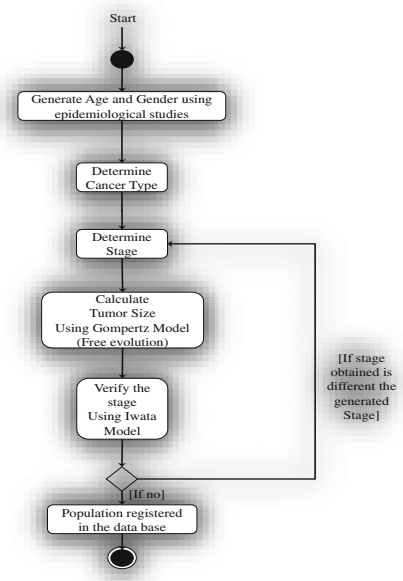


Fig. 5. Generate population using the epidemiological studies.

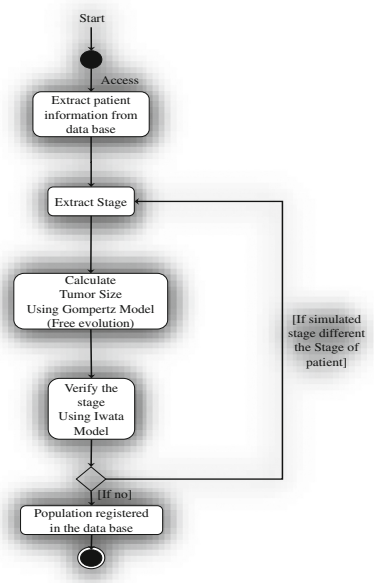


Fig. 6. Generate population using the real data.

6.2 Treatment Selection

Figure 7 shows a state chart for the treatment class which is responsible for generating a treatment plan for each patient. This treatment is defined by the physician. The patient has to perform some diagnosis which enables treatment plan choosing. When the

patient approves the treatment, the following information must be stored in the generated file to be used in the simulation step. In case the patient rejects the treatment, the physician has to choose another type of treatment in collaboration with the medical team and patient.

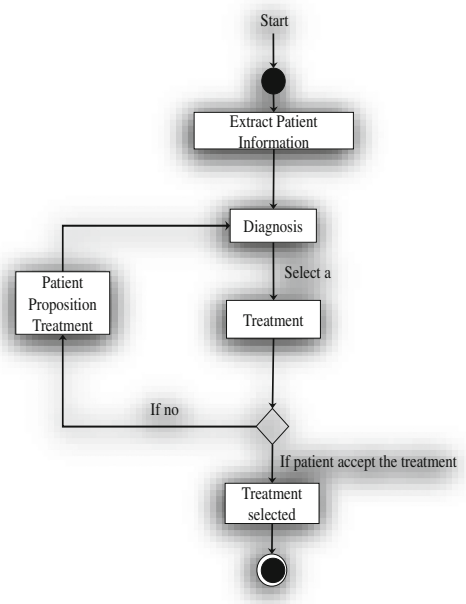


Fig. 7. Select treatment plan.

6.3 Treatment Trajectory

The Fig. 8 shows a state chart of class used to treat patients who have colon or rectal cancer. This treatment is created by the physician. First of all, the patient should be examined prior each treatment or session of treatments such as radiotherapy or chemotherapy. This is needed to evaluate the physical and psychological state of this patient and determine the stage of the cancer. After this evaluation the physician will be able to verify the patient's ability to continue treatment or suspend it for some period until the patients' state is re-evaluated or the treatment is adjusted (for example change the dose of the medication). During treatment, the patient may need to undergo more examinations if it is necessary or if the physician has any concerns.

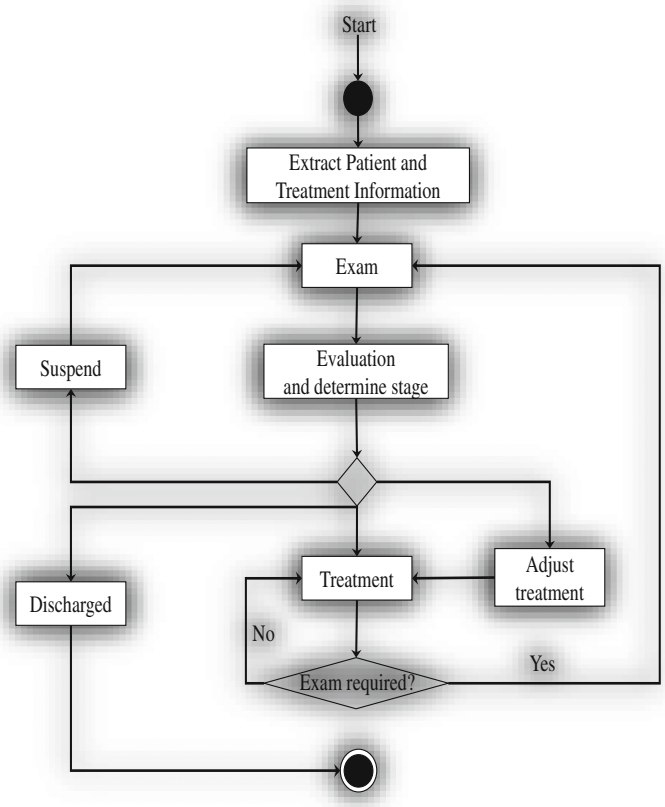


Fig. 8. Treatment phase.

6.4 Treatment Modifications During Chemotherapy

Treatment interruption or dose reduction was specified for reactions unlikely to become serious or life-threatening (National Cancer Institute of Common Toxicity Criteria) (Fig. 9). No dose was required for the grade 0, 1 and first appearance of a grade 2 toxicity. Treatment with capecitabine was interrupted in cases of grade 2 toxicity and was not continued until toxicity resolved or improved to grade 1. When treatment was continued, capecitabine doses were reduced as follows: (1) by 25% for patients who experienced a second appearance of a given grade 2 toxicity or any appearance of grade 3 toxicity or (2) by 50% for patients who experienced a third appearance of a given grade 2 toxicity, a second appearance of a given grade 3 toxicity, or any appearance of grade 4 toxicity. Treatment was discontinued if a given toxicity occurred, despite dose reduction, for a fourth time at grade 2, a third time at grade 3, or a second time at grade 4.

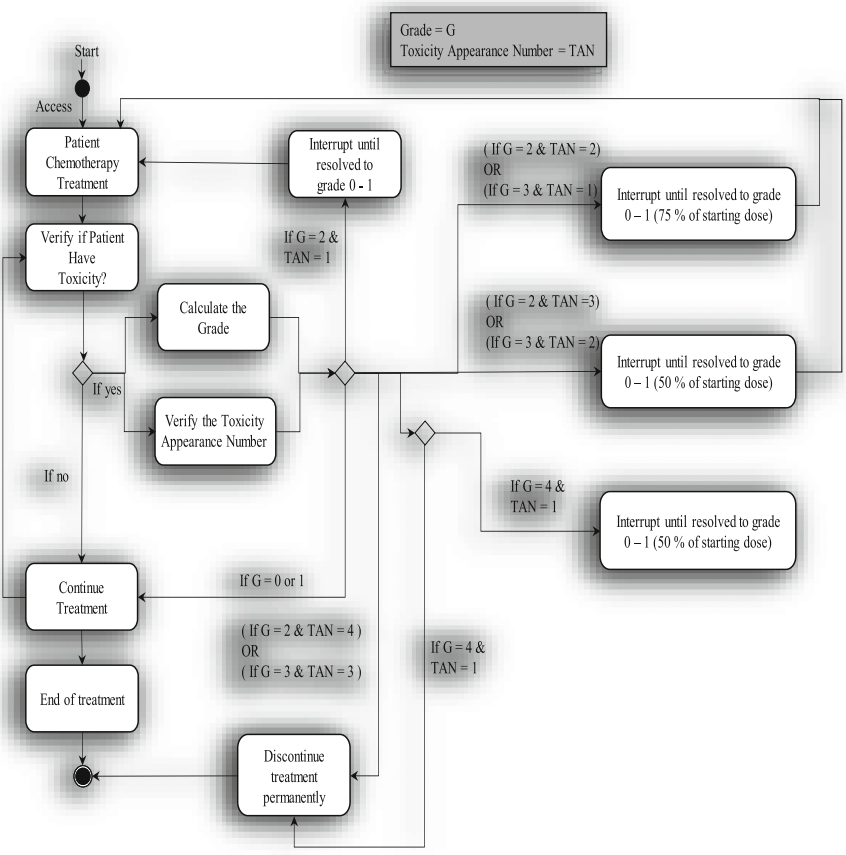


Fig. 9. Dose modification for capecitabine.

7 Validation

The first objective was to validate our modelling and simulation oriented agents; the results of the simulation are presented in Fig. 12. These simulations should allow us to validate our simulation platform for executing further simulations that involve treating patients with colon and rectal cancer. The input data of the simulation and the results are stored in a database, which was added into our simulation platform.

To do that, we carried out different experiments, using the Java eclipse software package with a 3,5 GHz Intel Core i7 processor and 32 GB of RAM. More specific, we used the JADE platform (Java Agent Development Framework). JADE it's a MAS development environment complies with the FIPA very diffused and included a set of

tools included facilitating various MAS development phases [31]. The experiment aims at assessing the ability of the model to replicate the results of real studies with specific treatment protocols. In order to compare the simulation results with actual data, we used the results for the real patients after treatments. The treatments results are classified by survived or not.

7.1 Experiment and Generate the Virtual Population

In this experiment, we must calibrate the model's parameters. In order to do this ABS, we use the Jewish Hospital real data, which allows us to validate our model during the different type of treatment. The real data include 773 patients who have colon and colorectal cancer. However, among these patients there are just 56 patients that have a complete profile, more precisely they have the stage, type of treatments, and the results after treatments which characterized by survived or not. Each of these patients have different types of information (or different profile), like stage, age, cancer type, type of treatment, and the protocol received by patients for different treatments. Patients in this protocol received two daily doses of chemotherapy treatment continuously without rest periods.

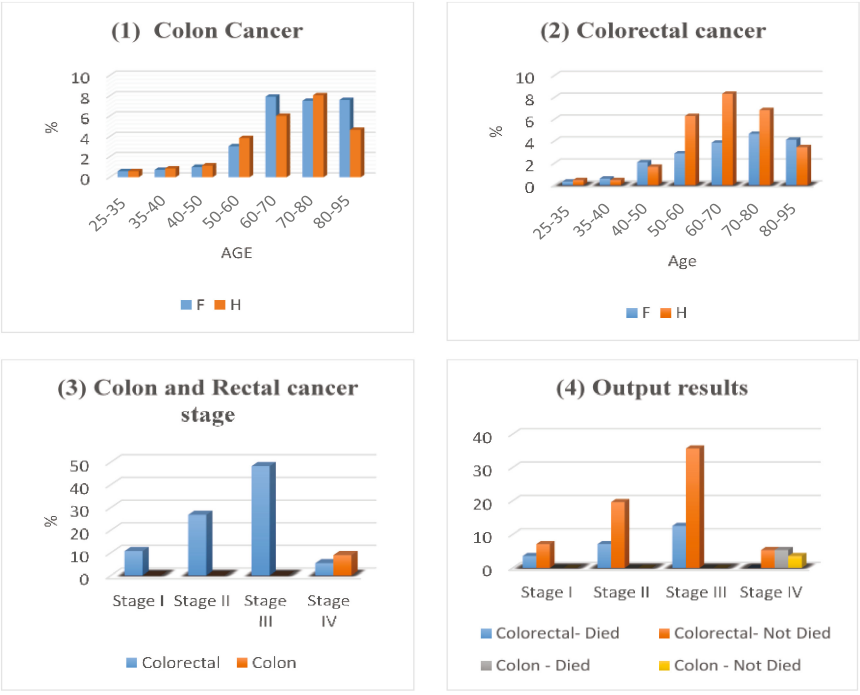


Fig. 10. Real data analyses.

Figure 10 explains the contents presented in the Jewish Hospital real data. Firstly, we present the percentage of patients (male and female) who have Colon and Rectal Cancer in our real data, secondly we present just the percentage of patients who have a colon and rectal cancer with complete profile, and finally we present the percentage of output results which is classified by Survived or not.

To start our simulation, we must create several populations of virtual patients based on the Jewish Hospital real data. Firstly, the user can be select one real patient from the data base to generate the following 100 virtual patients (or more) who have a similar real patient profiles and same treatment plan. During the generation of patient profiles, we used the Gompertz model which describes the evolution of the main tumour from the appearance of the first cancerous cell to a larger tumour (we determine the stage and the tumour size of the population generation). Then, we use the iwata model to describe the evolution of metastases. In case the stage obtained and real patient stages are matching, the population generated is registered in the data base and we can prepare the simulation step. However, we must generate the various parameters used by mathematics equations as Gompertz model and iwata model. In this chapter, the model was calibrated for one protocol.

To validate our work, we select five patients with different profiles to show the effectiveness of our model. The table below represents the Jewish Hospital (patients selected) and the population generation. This table presents the patient Number (P), the stage (St), the type of cancer (CT: Colon (C) or Colorectal (Col)), the different treatments (like surgery (Sug), Radiotherapy (Rad) and Chemotherapy (Ch)) and the Output results (Op) (Survived (S), or Not Survived (NS)). However, the real data and the population generation have the same profile (Table 1).

Table 1. Real data and population generation information.

	Jewish hospital real data							Population generation					
	PN	St	CT	Sug	Rad	Ch	Op	PN	S	CT	Sug	Rad	Ch
Experiment 1	1	II	Col	No	No	Yes	S	100	II	Col	No	No	Yes
Experiment 2	1	IV	Col	No	No	Yes	NS	100	IV	Col	No	No	Yes
Experiment 3	1	II	Col	No	No	Yes	S	100	II	Col	No	No	Yes
Experiment 4	1	III	Col	Yes	No	Yes	S	100	III	Col	Yes	No	Yes
Experiment 5	1	IV	C	Yes	No	No	S	100	IV	C	Yes	No	No

7.2 Calibration and Simulation Results

In order to calibrate the model for the configuration of the real hospital data, we first need to estimate the impact of each parameter on the results based on their role in the model. For example, the percentage of progressive disease is only defined by the parameter of the Gompertz evolution, the parameters of the chemotherapy E0, and Absorption and Dose. There are other parameters such as m, α and the maximum and minimum size of the tumour in the selection of the virtual population [9]. Thus, to calibrate the model, we proceed by trial-and-error, using a dichotomy approach to set each parameter and replicate the results of the hospital data as best as possible. Concerning the duration of the simulated treatments, the median duration reported in both studies was used for the corresponding tests. The final values of the parameters for each calibration are shown in Table 2. Concerning the parameter Absorption, it has been set equal to its defined in [36] value, while the average value of $\alpha 2$ was taken in [17].

Table 2. Calibration parameters.

	Experiment				
	1	2	3	4	5
Parameters	**Average (Standard deviation)**	**Average (Standard deviation)**	**Average (Standard deviation)**	**Average (Standard deviation)**	**Average (Standard deviation)**
m	$6*10^{-8}$ ($3*10^{-9}$)	$6*10^{-8}$ ($3*10^{-9}$)	$6*10^{-8}$ ($3*10^{-9}$)	$6*10^{-8}$ ($3*10^{-9}$)	$6*10^{-8}$ ($3*10^{-9}$)
α2	0,66 (0,03)	0,66 (0,03)	0,66 (0,03)	0,66 (0,03)	0,66 (0,03)
P	0.87	0.87	0.87	0.87	0.87
E0	$3,1*10^{-3}$	$3,1*10^{-3}$	$3,1*10^{-3}$	$3,1*10^{-3}$	$3,1*10^{-3}$
Absorption	0,6	0,6	0,6	0,6	0,6
Dose	> 0.45	> 0.45	> 0.45	> 0.45	> 0.45

Figures below show the stage before and after treatment for each experiments presented (before and after simulation). However, firstly we calculate the stage before start the treatment to precise a treatment plan for each patient, secondly, during the simulation step we need to verify the stage during the treatment in some cases to change the type of treatment/or change the radiotherapy and chemotherapy doses for example (Fig. 11).

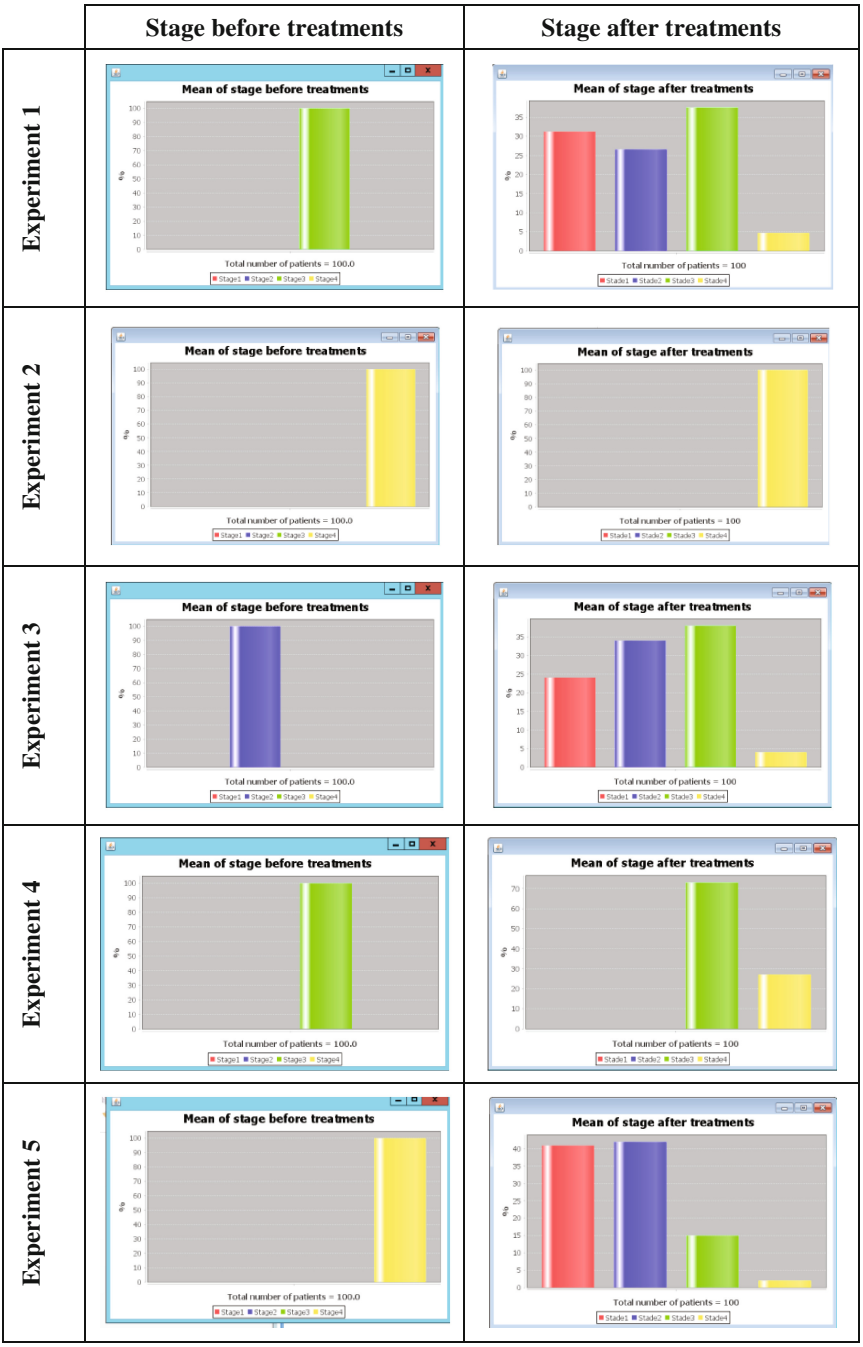


Fig. 11. Simulation results.

In figure below we compare our simulation results with the results presented in the real hospital data which are classified by survived or not.

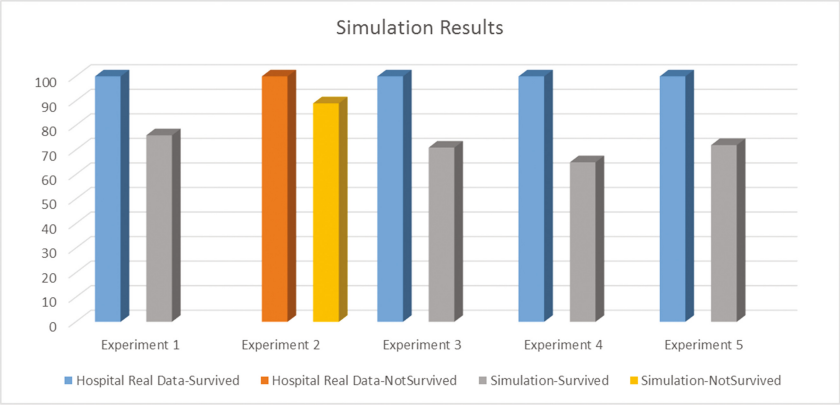


Fig. 12. Comparative analysis.

8 Discussion

The results obtained from the validation model tests are very interesting. In the specific context of some treatment (drug, dose and specific protocol), the model can be calibrated relatively easy to obtain excellent accuracy. Correspondingly, some model parameters can be properly adjusted in order to properly extrapolate the impact of a variable dose for a drug and a given protocol. So, in the agent based simulation platform, it is necessary to present a set of specific parameters for each drug, each protocol and duration of treatment, this implies a number of performing calibration. Moreover, further validation with specific data for each patient will be required to refine accuracy. In addition, several other aspects of the agent based conceptual model will be developed, such as the inclusion of side effects, etc. Similarly, the hospital's resources and management processes will also be modeled.

9 Conclusion and Future Work

This chapter introduced a conceptual model aiming at the development of a simulation environment capable of emulating the simultaneous care trajectories of several of cancer patients. Our research has focused on the definition of a modelling approach for agent's oriented simulation of HC, with the main objective to allow a more organizational modelling/agents oriented simulation of HC.

Before this model can be implemented and tested within the simulation environment, several other aspects of the conceptual model presented in Fig. 1 will have to be developed. Along the same line, the hospital resources processes will have to be modeled as well.

For this we have developed a simulation platform for the implementation of the conceptual model and implementation of multi-agent system. This platform used a simulation platform based on a specific simulation environment (JADE). This simulation allowed us to analyse the presented simulation behaviour in the HC system. We have conducted with our simulation platform several simulations of the HC allowing the study of several relevant scenarios.

The validation phase described in this chapter gives very important results to reality reproduce, but it is preliminary. Indeed, validation must be detailed with more specific data for each patient and have a better model calibrated than just on population averages, before integration in the simulation platform. Thus, validation with a more specific method, reflecting the better use of the model in the simulation platform, is required. This requires much more detailed data in the treatment of each patient, to be provided by the Jewish General Hospital in Montreal.

To complete the simulation platform, it will take the next step in this focus on the most important part will be the "Patient health model", because it will determine the impact of patient treatment side effects that is an important aspect of treatment against cancer. Indeed, the fight against cancer advanced by chemotherapy can be seen as a balance between enough drugs for reducing cancer, but not too much to not kill the patient.

References

1. Barbolosi, D., Verga, F., You, B., Benabdallah, A., Hubert, F., Mercier, C.: Modélisation du risque d'évolution métastatique chez les patients supposés avoir une maladie localisée. Oncologie **13**, 528–533 (2011)
2. Bolin, S., Nilsson, E., Sjödahl, R.: Carcinoma of the colon and rectum–growth rate. Ann. Surg. **198**, 151 (1983)
3. Chao, S., Wong, F.: A multi-agent learning paradigm for medical data mining diagnostic workbench (2009)
4. Claret, L., Girard, P., Hoff, P.M., Van Cutsem, E., Zuideveld, K.P., Jorga, K., et al.: Model-based prediction of phase III overall survival in colorectal cancer on the basis of phase II tumor dynamics. J. Clin. Oncol. **27**, 4103–4108 (2009)
5. Devi, M.S., Mago, V.: Multi-agent model for Indian rural health care. Leadersh. Health Serv. **18**, 1–11 (2005)
6. Foster, D., McGregor, C., El-Masri, S.: A survey of agent-based intelligent decision support systems to support clinical management and research. In: Proceedings of the 2nd International Workshop on Multi-agent Systems for Medicine, Computational Biology, and Bioinformatics, pp. 16–34 (2005)
7. Figueredo, G.P., Aickelin, U.: Comparing system dynamics and agent-based simulation for tumour growth and its interactions with effector cells. In: Proceedings of the 2011 Summer Computer Simulation Conference, pp. 52–59 (2011)
8. Fujimoto, R.: Parallel and Distributed Simulation Systems. Wiley, Hoboken (2000)
9. Gilli, Q., Mustapha, K., Frayret, J.M., Lahrichi, N., Karimi, E.: Agent-Based Simulation of Colorectal Cancer Care Trajectory: Patient Model. CIRRELT (Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation) (2014)

10. Gupta, S., Sarkar, A., Pramanik, I., Mukherjee, B.: Implementation scheme for online medical diagnosis system using multi agent system with JADE. Int. J. Sci. Res. Publ. **2**(6) (2012). ISSN 2250-3153
11. Gyllenberg, M., Webb, G.F.: Quiescence as an explanation of Gompertzian tumor growth. Growth Dev. Aging: GDA **53**, 25–33 (1988)
12. Gupta, S., Mukhopadhyay, S.: Multi agent system based clinical diagnosis system: an algorithmic approach. Int. J. Eng. Res. Appl. (IJERA) **2**(5), 1474–1477 (2012). ISSN: 2H8-9622
13. Gupta, S., Pujari, S.: A multi-aged: based scheme for heath care and clinical diagnosis system. In: IAMA (2009). IEEExplore. ISBN 978-1-4244-4710-7
14. Han, B.-M., Song, S.-J., Lee, K.M., Kyung-Soo, J., Dong-Ryeol, S.: Multi agent system based efficient healthcare service. In: ICACT 2006, 20–24 February 2006. ISBN 89-551 9-1 29-4
15. Heun, J.M., Grothey, A., Branda, M.E., Goldberg, R.M., Sargent, D.J.: Tumor status at 12 weeks predicts survival in advanced colorectal cancer: findings from NCCTG N9741. Oncologist **16**, 859–867 (2011)
16. Iantovics, B.: The CMDS medical diagnosis system. In: Ninth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. IEEE (2008)
17. Iwata, K., Kawasaki, K., Shigesada, N.: A dynamical model for the growth and size distribution of multiple metastatic tumors. J. Theor. Biol. **203**, 177–186 (2000)
18. Jennings, N.R., Sycara, K., Wooldridge, M.: A roadmap of agent research and development. Auton. Agent. Multi-agent Syst. **1**, 7–38 (1998)
19. Jones, S.S., Evans, R.S.: An agent based simulation tool for scheduling emergency department physicians. In: AMIA Annual Symposium Proceedings, p. 338 (2008)
20. Kanagarajah, A., Parker, D., Xu, H.: Health care supply networks in tightly and loosely coupled structures: exploration using agent-based modelling. Int. J. Syst. Sci. **41**, 261–270 (2010)
21. Kazar, O., Sahnoun, Z., Frecon, L.: Multi-agents system for medical diagnosis. In: International Conference on Intelligent System and Knowledge Engineering, vol. 1, pp. 1265–1270 (2008)
22. Klusch, M., Lodi, S., Moro, G.: The role of agents in distributed data mining - issues and benefits. In: Proceedings of IEEE/WIC International Conference on Intelligent Agent Technology (IAT 2003), pp. 211–217 (2003)
23. Knight, V.A., Williams, J.E., Reynolds, I.: Modelling patient choice in healthcare systems: development and application of a discrete event simulation with agent-based decision making. J. Simul. **6**, 92–102 (2012)
24. Krizmaric, M., Zmauc, T., Micetic-Turk, D., Stiglic, G., Kokol, P.: Time allocation simulation model of clean and dirty pathways in hospital environment. In: Proceedings of 18th IEEE Symposium on Computer-Based Medical Systems, pp. 123–127 (2005)
25. Laskowski, M., McLeod, R.D., Friesen, M.R., Podaima, B.W., Alfa, A.S.: Models of emergency departments for reducing patient waiting times. PLoS ONE **4**, e6127 (2009)
26. Mahmud, R., Sithiq, H.A.A.H., Taharim, H.M.: A Hybrid Technology for a Multi agent Consultation System in Obesity Domain. World Academy of Science, Engineering and Technology (2009)
27. Mustafee, N., Katsaliaki, K., Taylor, S.J.E.: Profiling literature in healthcare simulation. Simulation **86**, 543–558 (2010)
28. Nealon, J., Moreno, A.: Agent-based applications in health care. In: Applications of Software Agent Technology in the Health Care Domain, pp. 3–18. Springer (2003)

29. Quesnel, G., Duboz, R., Ramat, E., Traoré, M.K.: VLE: a multimodeling and simulation environment. In: Proceedings of the Summer Simulation Multiconference (SummerSim 2007), San Diego, California, USA, pp. 367–374, 15–18 July 2007
30. Rosa, M.V., Flores, C.D., Silvestre, A.M., Seixas, L.J., Ladeira, M., Coelho, H.: A multi agent intelligent environment for medical knowledge. Artif. Intell. Med. **27**, 335–366 (2003)
31. Rimassa, G., Bellifemine, F., Poggi, A.: JADE - a FIPA compliant agent framework. In: PMAA 1999, Londres, pp. 97–108 (1999)
32. Canadian Cancer Society: Subcutaneous port. http://www.cancer.ca/en/cancer-information/diagnosis-and-treatment/tests-and-procedures/subcutaneous-port/?region=on
33. Canadian Cancer Society: Peripherally inserted central catheter. http://www.cancer.ca/en/cancer-information/diagnosis-and-treatment/tests-and-procedures/peripherally-inserted-central-catheter/?region=on
34. Stainsby, H., Taboada, M., Luque, E.: Towards an agent-based simulation of hospital emergency departments. In: SCC IEEE International Conference on Services Computing, Bangalore, India, 21–25 September 2009, pp. 536–539 (2009)
35. Suwinski, R., Wzietek, I., Tarnawski, R., Namysl-Kaletka, A., Kryj, M., Chmielarz, A.: Moderately low alpha/beta ratio for rectal cancer may best explain the outcome of three fractionation schedules of preoperative radiotherapy. Int. J. Radiat. Oncol. Biol. Phys. **69**, 793–799 (2007)
36. Verga, F.: Modélisation mathématique de processus métastatiques, Université de Provence-Aix-Marseille I (2010)
37. Van Cutsem, E., Findlay, M., Osterwalder, B., Kocha, W., Dalley, D., Pazdur, R., et al.: Capecitabine, an oral fluoropyrimidine carbamate with substantial activity in advanced colorectal cancer: results of a randomized phase II study. J. Clin. Oncol. **18**, 1337–1345 (2000)
38. Wang, P., Feng, Y.: A mathematical model of tumor volume changes during radiotherapy. Sci. World J. **2013** (2013)
39. Zhang, W., Yao, Z.: A reformed lattice gas model and its application in the simulation of evacuation in hospital fire. In: IEEE International Conference on Industrial Engineering and Engineering Management, IEEM 2010, Macao, China, pp. 1543–1547, 7 December 2010
40. Zhang, C., Cao, L.: Agents and data mining: mutual enhancement by integration. autonomous. In: Gorodetsky, V., Liu, J., Skormin, V.A. (eds.) Autonomous Intelligent Systems: Agents and Data Mining, pp. 50–61. Springer, Heidelberg (2005)

Parameter Identification of Canalyzing and Nested Canalyzing Boolean Functions with Ternary Vectors for Gene Networks

Annika Eichler[1] and Gerwald Lichtenberg[2(✉)]

[1] Automatic Control Laboratory, ETH Zurich, Physikstrasse, Zurich, Switzerland
[2] Faculty Life Sciences, Hamburg University of Applied Sciences, Ulmenliet, Hamburg, Germany
Gerwald.Lichtenberg@haw-hamburg.de

Abstract. In gene dynamics modeling, parameters of Boolean networks are identified from continuous data under various assumptions expressed by logical constraints. These constraints may restrict the dynamics of the network to the subclass of canalyzing or nested canalyzing functions, which are known to be appropriate for genetic networks. This paper introduces high performance algorithms, which solve the parameter identification problem by so called Zhegalkin identification and exploit the restriction to canalyzing or nested canalyzing functions resulting in reduced calculation time. The constraints are formulated in terms of orthogonal ternary vector lists, which offer an efficient representation for Boolean functions. The canalyzing constraints can be intrinsically incorporated in an existing Branch-and-Cut algorithm, which lead to a natural restriction of the search space and thus of the calculation time. For nested canalyzing constraints this is not possible. Instead, an identification algorithm based on enumeration is proposed. The algorithms are applied to mRNA micro array data from mice under different contaminant conditions and good correspondence to a known apoptotic pathway can be shown.

Keywords: Parameter identification · Networks · Gene dynamics · Systems biology · Boolean functions · Ternary logic

1 Introduction

A current field of research in systems biology is gene dynamics modeling, since understanding the dynamics of the genetic model could help the therapeutic process [17].

Kaufman [11] has shown that Boolean function are appropriate to model genetic networks, due to their common characteristics, as periodicity, global complexity and self organization. Canalyzing functions, introduced by Kauffman in 1993, are a subclass of Boolean functions, which turned out to describe the

M.S. Obaidat et al. (eds.), *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing 676,
https://doi.org/10.1007/978-3-319-69832-8_12

highly ordered dynamics of gene networks better than other Boolean models, due to their stabilizing effect on the discrete dynamical behavior [4,12,13]. In genetic networks canalization is the ability of a genotype to produce the same phenotype regardless of environmental variability [8]. With nested canalyzing functions [11], Kauffman et al. introduced in 2003 a subclass of canalyzing function with increased stabilizing effects. These seem to be particularly capable to describe genetic networks [8].

A successful approach to identify parameters of Boolean functions from contiuous-valued signals like microarray data uses Zhegalkin polynomials to represent these functions, see [2,5,16,20]. The Zhegalkin identification problem is a Mixed Integer Quadratic Program (MIQP) which can in principle be solved with standard tools like CPLEX or Xpress, where Branch-and-Cut algorithms are used. One major problem of Boolean identification is the exponential growth of the cardinality of the solution set with the number of interacting genes. Thus, those methods are applicable up to a model order of $n = 10$, where already very large runtimes of hours or days occur, [4].

Furthermore, a clustering problem has to be solved to determine groups of genes of unknown cardinality—denoted *connectivity degree*—which affect each other. Combining the clustering and the Zhegalkin identification problem leads to a problem of discrete optimization with even higher complexity. First approximations for the solution of this combined problem have been found by a preprocessing step based on the Pearson Correlation Coefficient in [4]. Next, exploiting efficient representations of Zhegalkin polynomials as orthogonal ternary vector lists (OTVLs), [1], and adapting tensor decomposition techniques from [14] allows integration of both steps reported in [15]. Moreover, the solution set of the identification algorithm can be reduced by fixing the maximum number of rows of the OTVL representing the solution. This leads to highly efficient computation with controllable degree of accuracy, because optimality of the solution is guaranteed by a Branch-and-Cut algorithm used for the reduced solution set. In this paper, the latter method is restricted to the subclass of canalyzing functions (CFs) due to their interesting properties. This introduces additional constraints for the optimization problem, as already reported in [2,7], but the reduced solution set is not efficiently exploited therein. This work shows how to incorporate those constraints in the Branch-and-Cut identification algorithm in [15] by expressing canalizing functions as OTVLs based on [3]. The proposed algorithm for the identification of CFs is by orders of magnitude more efficient since the search space is considerably reduced as obvious from Table 1. Furthermore, it is shown how to express nested canalyzing functions (NCFs) as OTVLs to exploit their efficient representation. A simple incorporation of NCFs in the Branch-and-Cut algorithm is not possible due to their iterative definition. Instead, an identification algorithm via enumeration is proposed, which is up to a reasonable connectivity degree computationally tractable and favorable compared to the Branch-and-Cut algorithm without constraints because of their small number. The constrained identification methods are applied to gene expression data from mRNA extracted from mouse liver cells.

This work is organized as follows. Section 2 introduces fundamentals of Boolean functions, Zhegalkin polynomials and OTVLs. In Sect. 3 the Branch-and-Cut Boolean identification algorithm from [15] is described. Section 4 presents how to express CFs as OTVLs and adapt the identification therefore. In Sect. 5 the formulation of NCFs as OTVLs is described. The results on an application to real data are shown in Sect. 6. Finally conclusion are drawn in Sect. 7.

2 Fundamentals

The set $\mathbb{B}=\{0,1\}$ denotes the set of logicals, $\mathbb{U}=[0,1]$ the unit interval. Negation of Booleans is denoted by $\neg z=\bar{z}$, for real variables $\bar{x}=1-x$ holds. With $\otimes$ the Kronecker product is denoted.

2.1 Boolean Functions and Zhegalkin Polynomials

A Boolean function (BF) $b:\mathbb{B}^n\to\mathbb{B}$ can be represented by its truth vector $\mathbf{b}=(b_1,...,b_{2^n})'\in\mathbb{B}^{2^n}$, i.e., the last column of the truth table as shown in Table 2.

Example 1 ([3]). Consider the Boolean function

$$b(y_1,y_2)=\neg(y_1\wedge y_2), \tag{1}$$

which is given by the truth table

y_2	y_1	$\mathbf{b}(y_1,y_2)$
0	0	1
0	1	1
1	0	1
1	1	0

(2)

with its truth vector $\mathbf{b}=\begin{pmatrix}1&1&1&0\end{pmatrix}'$.

Definition 1. *A* Zhegalkin polynomial $p(\mathbf{y})=\mathbf{l}(\mathbf{y})'\mathbf{b}$ *is a multilinear polynomial with* $\mathbf{b}\in\mathbb{B}^{2^n}$ *being a truth vector and* $\mathbf{l}(\mathbf{y})$ *the so called* literal vector*, given by [15] as*

$$\mathbf{l}(\mathbf{y})=\begin{pmatrix}\bar{y}_n\\ y_n\end{pmatrix}\otimes\cdots\otimes\begin{pmatrix}\bar{y}_1\\ y_1\end{pmatrix}\in\mathbb{U}^{2^n}. \tag{3}$$

Proposition 1 ([21]). *A Zhegalkin polynomial evaluated at Boolean values* $\mathbf{y}\in\mathbb{B}^n$ *gives the same (Boolean) result as the BF represented by the truth vector* $\mathbf{b}$*.*

Thus the Zhegalkin polynomials can be seen as the bridge between the Boolean and the real set $\mathbb{U}$. Since if $\mathbf{y}\in\mathbb{U}$ then $p(\mathbf{y})\in\mathbb{U}$ as well, if however $\mathbf{y}\in\mathbb{B}$ then $p(\mathbf{y})\in\mathbb{B}$.

Table 1. Number of BFs, CFs and NCFs.

n	BFs	CFs	NCFs
1	4	4	2
2	16	14	8
3	256	120	64
4	65536	3514	736
5	$4.2950 \cdot 10^9$	1292276	10624
6	$1.8447 \cdot 10^{19}$	$1.0307 \cdot 10^{11}$	183936

Table 2. Truth table.

y_n	$\cdots$	y_2	y_1	$\mathbf{b}(y_1, ..., y_n)$
0	$\cdots$	0	0	b_1
0	$\cdots$	0	1	b_2
0	$\cdots$	1	0	b_3
0	$\cdots$	1	1	b_4
$\vdots$		$\vdots$	$\vdots$	$\vdots$
1	$\cdots$	1	1	b_{2^n}

Example 1 (continued). To illustrate this for the BF (1) the corresponding Zhegalkin polynomial is calculated as

$$\mathbf{l}'(\mathbf{y})\mathbf{b} = \begin{pmatrix} (1-y_1)(1-y_2) \\ y_1(1-y_2) \\ (1-y_1)y_2 \\ y_1 y_2 \end{pmatrix}' \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} = 1 - y_1 y_2. \tag{4}$$

It can be easily seen that if $y_1, y_2 \in \mathbb{B}$, then the Zhegalkin polynomial leads to the same solution as the BF (1), as declared in Proposition 1.

2.2 Ternary Vector Lists

Ternary Vector Lists (TVLs) are a common concept in Boolean algebra, because of its outstanding advantages for large scale problems, [1]. A TVL of a BF represents all elements of the Boolean space $\mathbb{B}^{2^n}$ where the function is 1 by ternary vectors (TVs). A TV $\mathbf{t}$ has the structure

$$\mathbf{t} \in \mathbb{T}^n = \{0, 1, -\}^n . \tag{5}$$

A zero element '0' in the TV describes that the corresponding variable appears negated, a one element '1' that it appears not negated. The latter '$-$' is the *don't care* symbol, that can stand for either '1' or '0'.

A TVL with k lines is of the form

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_k \end{bmatrix} .$$

Taking all lines of the truth table with ones always leads to a valid TVL of a BF. TVLs with smaller number of lines might be possible by using '$-$'.

Example 1 (continued). With the truth table in (2) valid TVLs for the BF (1) of the running example are

$$\mathbf{T_1} = \begin{bmatrix} 00 \\ 01 \\ 10 \end{bmatrix}, \quad \mathbf{T}_2 = \begin{bmatrix} 01 \\ -0 \end{bmatrix}, \quad \mathbf{T}_3 = \begin{bmatrix} 0- \\ -0 \end{bmatrix}, \quad \mathbf{T}_4 = \begin{bmatrix} 0- \\ 10 \end{bmatrix}. \tag{6}$$

This can easily be checked by replacing '−' with both '0' and '1'.

This example shows that TVLs are not unique, i.e., there exist different TVLs for the same BF. Another important property is orthogonality [1].

Definition 2. *A TVL* $\mathbf{T}$ *is orthogonal, if each binary vector appears only once in* $\mathbf{T}$*. This is the case, if for any pair of lines of* $\mathbf{T}$ *in at least one column a* (0,1)*-combination appears. Two TVLs* $\mathbf{T}_A$ *and* $\mathbf{T}_B$ *are orthogonal if* $\mathbf{T}_A$ *and* $\mathbf{T}_B$ *have no binary vectors in common. This is the case if for any pair of lines of* $\mathbf{T}_A$ *and* $\mathbf{T}_B$ *in at least one column a* (0,1)*-combination appears.*

A binary vector (BV) is a vector with only '0's and '1's. It can represent only one line of the truth table, while a ternary vector (TV) due to '−' can represent multiple BVs.

Example 1 (continued). For the TVLs of the example it is obvious that all TVL representations are orthogonal except of $\mathbf{T}_3$ with no (0,1)-combination in any column. Here the binary vector [00] appears in both lines.

In the following an orthogonal TVL is denoted as OTVL. In [1] operations for OTVLs are described. Important for this work are the complement and the difference operators, which are visualized in Table 3 for 3 variables. The complement $\mathrm{CPL}(\mathbf{T}) = \bar{\mathbf{T}}$ of a given OTVL $\mathbf{T}$ is defined as the OTVL of all binary vectors that are not in $\mathbf{T}$. The difference $\mathrm{DIF}(\mathbf{T}_A, \mathbf{T}_B)$ of the OTVLs $\mathbf{T}_A$ and $\mathbf{T}_B$ results in an OTVL of all BVs, that are in $\mathbf{T}_A$ but not in $\mathbf{T}_B$. If the result is the empty OTVL [ ], $\mathbf{T}_A$ is totally included in $\mathbf{T}_B$.

Table 3. Graphical representation of operands for TVLs, [1].

$\mathbf{T_A} = \begin{bmatrix} 00- \\ 1-1 \end{bmatrix}$ 	$\mathrm{CPL}(\mathbf{T}_A) = \bar{\mathbf{T}}_A = \begin{bmatrix} 01- \\ 1-0 \end{bmatrix}$ 
$\mathbf{T_B} = [1--]$ 	$\mathrm{DIF}(\mathbf{T_A},\mathbf{T_B}) = [00-]$ 

Lemma 1 ([3]). *An OTVL* $\mathbf{T}$ *is orthogonal to its complement* $\bar{\mathbf{T}}$*.*

Proof. With Definition 2 two TVLs are orthogonal, if they do not have any BVs in common. The complement of an OTVL $\mathbf{T}$ contains all BVs, that are not in $\mathbf{T}$ and is thus orthogonal to $\mathbf{T}$.

Proposition 2 ([3]). *For an OTVL* $\mathbf{T}$ *with* k *lines the number of ones in the corresponding truth vector* $\mathbf{b}$ *is* $N_1 = \mathbf{b}'\mathbf{1} = \sum_{i=1}^{k} 2^{N_{i-}}$ *where* N_{i-} *is the number of '−'s in the* i*-th line of* $\mathbf{T}$.

Proof. The number ones in $\mathbf{b}$ is equivalent to the number of BVs in $\mathbf{T}$. A TV with no '−'s represents a single BV and since a '−' stands for either `1` or `0`, a TV with N_- times the '−' symbol, includes 2^{N-} BVs. Due to orthogonality no BV appears more than once in $\mathbf{T}$, so that the number of BVs in each line can simply be added.

2.3 OTVLs and Zhegalkin Polynomials

Since OTVLs and Zhegalkin polynomials are two different representations of BFs, it is possible to find the corresponding mapping between both representations.

Proposition 3 ([3]). *Given is an OTVL* $\mathbf{T}$ *of* n *variables, that is representing a BF* f*, then the corresponding Zhegalkin polynomial, determined by* $p_{\mathbf{T}}$*, is calculated as*

$$p_{\mathbf{T}}(\mathbf{y}) = \sum_{j=1}^{k} \prod_{i=1}^{n} T(t_{ji}, y_i)$$

$$with\, T(t_{ji}, y_i) = \begin{cases} \bar{y}_i, & if\, t_{ji} = \mathtt{0}\,, \\ y_i, & if\, t_{ji} = \mathtt{1}\,, \\ 1, & if\, t_{ji} = -\,. \end{cases} \tag{7}$$

Proof. Assume $\mathbf{T}$ is an OTVL, i.e. without '−'s, then $\prod_{i=1}^{n} T(t_{ji}, y_i)$ corresponds to the l-th row of the literal vector. Since $\mathbf{t}_j$ is only a line of $\mathbf{T}$ when $\mathbf{b}_l = 1$ due to the construction of an OTVL, (7) is equal to $\mathbf{l}(\mathbf{y})'\mathbf{b}$, what finishes the proof for OTVLs without '−'s. If $\mathbf{T}$ is an OTVL with a '−' in the k-th column, than this is equal to a TVL with the same row and a '`1`' in the k-th column and additionally the same row and a '`0`' in the k-th column. For the row with the '`1`', if it is the n-th row, it is $\prod_{i=1}^{n} T(t_{ni}, y_i) = y_k \prod_{i=1, i\neq k}^{n} T(t_{ni}, y_i)$, and for that with the '`0`', if it is the m-th row, it is $\prod_{i=1}^{n} T(t_{mi}, y_i) = \bar{y}_k \prod_{i=1, i\neq k}^{n} T(t_{mi}, y_i)$. Thus the sum is $(y_k + \bar{y}_k) \prod_{i=1, i\neq k}^{n} T(t_{mi}, y_i) = \prod_{i=1, i\neq k}^{n} T(t_{mi}, y_i)$, since $\prod_{i=1, i\neq k}^{n} T(t_{mi}, y_i) = \prod_{i=1, i\neq k}^{n} T(t_{ni}, y_i)$. What finishes the proof for all OTVLs.

Example 1 (continued). Let's consider $\mathbf{T}_2 = \begin{bmatrix} \mathtt{01} \\ \mathtt{-0} \end{bmatrix}$ of the running example. Evaluating (7) for $\mathbf{T}_2$ leads to

$$p(\mathbf{y}) = \bar{y}_1 y_2 + 1\bar{y}_2 = (1 - y_1)y_2 + (1 - y_2) = 1 - y_1 y_2$$

as derived with the literal form (4) before.

3 Zhegalkin Identification by Branch-and-Cut Algorithm

Finding the best Boolean model for continuous normalized data is known as *Zhegalkin identification* problem, see [6], that has been shown to be well suited for Boolean identification of gene networks [2,4,20]. In [15] the Zhegalkin identification problem is solved with the help of OTVLs by a Branch-and-Cut algorithm.

In contrast to the first references, the efficient algorithm in [15] allows to include this clustering problem in the identification. A cluster is denoted as the set of genes, which affects the dynamics of a gene of interest, since a gene is never affected by all others genes, but only a subset, the cluster. The size of the cluster, called connectivity degree, and the cluster itself are unknown and have to be determined in the clustering problem. To build on the Zhegalkin identification algorithm from [15] it is shortly introduced here.

3.1 Minimization Problem

A Zhegalkin function of n signals can be modeled by n truth vectors or the respective OTVLs. The state space model for signal l is then given as

$$y_l(t+1) = \mathbf{l}(\mathbf{y}(t))'\mathbf{b}_l = p_{\mathbf{T}_l}(\mathbf{y}(t)), \qquad \forall l = 1,\dots,n, \tag{8}$$

with $p_{\mathbf{T}_l}(\mathbf{y})$ as defined in (7). The prediction error between $y_l(t+1)$ predicted with the OTVL $\mathbf{T}_l$ as model as in (8) and the measurement value $\tilde{y}_l(t+1)$ of signal l at any time $t = 0, \cdots, T-1$ is defined as $d_l(t+1) = y_l(t+1) - \tilde{y}(t+1)$. The task of the Zhegalkin identification problem is to find the optimal OTVL $\mathbf{T}_l^\star$ and the corresponding Zhegalkin polynomial that solves the minimization problem

$$\min_{\mathbf{T}_l} J_l \quad \text{with} \quad J_l = \sqrt{\sum_{t=0}^{T-1} d_l(t)^2}. \tag{9}$$

It is clear that this minimization problem has to be solved for all signals $l = 1,\dots,n$. Therefore this index is omitted in the following.

One major problem of Boolean and thus Zhegalkin identification is the high cardinality of the search space. There exist $2^{(2^n)}$ different BFs of n variables. This fast growth in the number of variables n is exemplarily shown in Table 1. To deal with this problem, the algorithm presented here finds the best approximation $\mathbf{T}^+$ with fixed maximal number of rows, instead of searching for the optimal solution. This row restriction significantly reduces the search space by preserving the basic properties as it is approved in Sect. 6 by the numerical example.

3.2 Branch-and-Cut Algorithm

The Zhegalkin identification with rank restriction from [15] is a Branch-and-Cut algorithm, where the nodes represent possible OTVLs. The algorithm is

initialized with the empty OTVL. The children in the next level are all 3^n OTVLs with one line. The following levels are built respectively by adding one TV, that is orthogonal to the parent node, to the OTVL of the parent node while descending in the search tree. This is equivalent to elongate the OTVL of the parent node by one line. The algorithm can be summarized in the following steps:

(1) Initialization
(2) Repeat: Define branching node, branch node, cut nodes
(3) End: According to stop criteria

The implemented Branch-and-Cut algorithm uses a best first strategy, therefore, the branching node is always the leaf (node without children) with smallest error function and with less than the maximal permitted row number. When branching, for each TV that is orthogonal to the OTVL of the branching node, a leaf, where this TV is added to it, is generated. For each new node the prediction error is calculated, and when it is clear, that this new branch can not decrease the current global best solution J^+, the node is cut, i.e., deleted from the search tree. The cutting condition hereby is

$$\text{cut node } j \text{ if } \exists t \in \{1, ..., T\} : d_j(t) > \hat{J}^+ . \tag{10}$$

Here $d_j(t)$ is the prediction error of node j at time t and $\hat{J}^+$ is the cost of the current best solution. The cutting condition (10) can be explained by the fact that $y(t) \in \mathbb{U}$ and thus non-negative. Therefore the Zhegalkin polynomial of every TV is non-negative as well. Thus if the modeled value for one time exceeds the measured one by more than the current error, the error can not get smaller if a further TV is added.

This is visualized for a simple example of $n = 4$ in Fig. 1. The blue points describe all nodes of the search tree. The straight red lines describe the cutting planes (cutting lines for $T = 2$), defined by $\hat{J}^+$, the distance of the current best solution and the measurement. All nodes outside the cutting planes are worse than the best solution and further branching of those nodes can not improve the solution, because by adding another TV something positive (or zero) in all directions is added. For more explanations see [15].

Several stopping criteria exist, like a desired lower threshold of the cost or a maximum number of iterations, can be set manually. If the algorithm stops because no node is branchable anymore, i.e., every leaf has reached the maximal permitted row number, then the optimal $\mathbf{T}^+$ in the restricted search space is found with minimal cost J^+.

Including the Clustering Problem. In general the Branch-and-Cut algorithms runs for each possible cluster, set of genes the considered gene may depend on, separately. However, if the initial lower bound $\hat{J}^+$ for each new cluster is set to the lowest optimal bound J^+ of all previously identified clusters, advantage of this information can be taken: if a cluster with a very good solution has been found, the cutting condition (10) of the following clusters is tightened from the beginning on, i.e., a lot of nodes are cut, leading to reduced calculation effort.

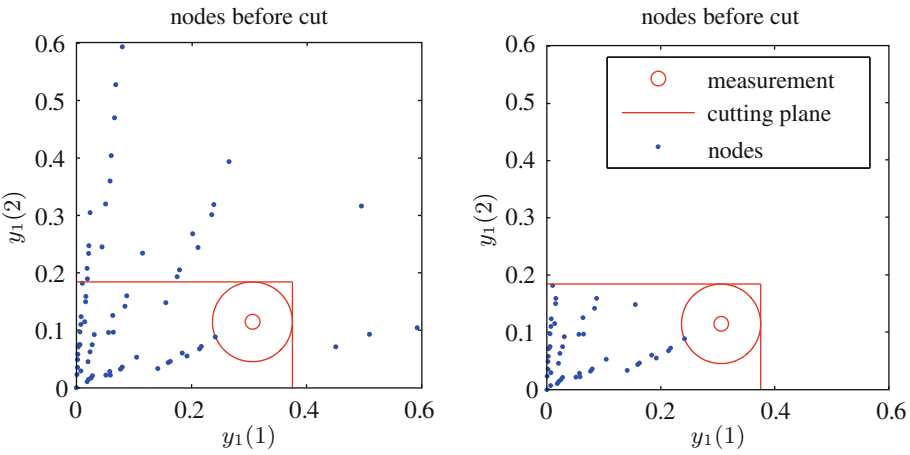


Fig. 1. Evaluating the cutting condition [15].

4 Canalyzing Functions

As mentioned above CFs and NCFs are subclasses of the BFs that are particularly suited to model genetic networks. Therefore, it is discussed in the following how to extend or adapt the Boolean identification algorithm, such that the resulting identified model is restricted to be either canalyzing or nested canalyzing. One possibility therefore would be to include a canalyzing or nested canalyzing test in each branching step. This however is only efficient if the test is computationally cheap to be realized and if the ratio between the number of BFs and those to be tested for is large enough. Since both points do not hold for CFs nor for NCFs, testing would not be efficient. Alternative solutions are searched for instead, starting with CFs. For these, it is shown in the following how to adapt the root of the Branch-and-Cut search tree, such that only CFs can be generated by branching. This is very efficient since no testing is needed. Furthermore, the rank restriction can be applied as before to further reduce the search space and save calculation time. In the next section it is shown that this is not as easy possible for NCFs.

4.1 Definition of Canalyzing Functions

CFs are a subclass of BFs with the property, that their result is fixed, if one specific input takes a specific value, no matter what values the other inputs take.

Definition 3 ([16]). *A Boolean function f is canalyzing if there exists an $i \in \{1,...,n\}$ and a fixed s, $v \in \{0,1\}$ such that for all $y \in \mathbb{B}^n$ we have $f(y_1,...,y_i,...,y_n) = v$ if $y_i = s$.*

The variable y_i is termed as *canalyzing variable*, s as *canalyzing value* and v as *canalyzed value*. If no i can be found, so that the condition above is fulfilled, the function is classified as non-canalyzing. For a canalyzing Boolean function the following holds

Lemma 2 ([3]). *Given a BF f for n variables that is canalyzing in y_i with canalyzing value s and canalyzed value v, then its complement $\bar{f}$ is canalyzing in y_i with s and $\bar{v}$.*

Proof. The complement of the BF f is defined as $\bar{f} = 1 - f$. Thus if $f(y_1, ..., y_i = s, ..., y_n) = v$ the complement $\bar{f}$ evaluated for $y_i = s$ is

$$\bar{f}(y_1, ..., y_i = s, ..., y_n) = 1 - v = \bar{v}\,.$$

In total, according to [9] the number of CFs for n variables is

$$N_c(n) = 2\left((-1)^n - n\right) + \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} 2^{k+1} 2^{2^{n-k}}\,.$$

The values of $N_c(n)$ for $n = 1, ..., 6$ are shown in Table 1.

4.2 OTVLs of Canalyzing Functions

Whereas expressing canalyzing functions as Zhegalkin polynomials has been considered in [4,5], this work is focused on expressing canalyzing in form of OTVLs to be able to restrict the Branch-and-Cut algorithm of Sect. 3 to only canalyzing functions.

If a BF is canalyzing, for the respective OTVL one of the two following Lemmas holds, depending on the canalyzed value.

Lemma 3 ([3]). *Given an OTVL $\mathbf{T}$ for n variables and with k lines, then $\mathbf{T}$ is canalyzing in variable y_c with canalyzing value s and canalyzed value $v = 0$ if and only if $t_{jc} = \bar{s}$ for all $j = 1, \ldots, k$.*

Proof. The corresponding Zhegalkin polynomial is calculated by (7). Since $t_{jc} = \bar{s}$ for all $j = 1, \ldots, k$, (7) can be written as

$$p_{\mathbf{T}}(\mathbf{y}) = T(\bar{s}, y_c) \sum_{j=1}^{k} \prod_{i=1, i \neq c}^{n} T(t_{ji}, y_i). \tag{11}$$

If $y_c = s$, i.e. the canalyzing value is taken, then $T(\bar{s}, y_c) = T(\bar{s}, s) = 0$, thus $p(\mathbf{y})$ with $y_c = s$ is equal to $v = 0$.

Lemma 4 ([3]). *Given an OTVL $\mathbf{T}$ for n variables and with k lines, then $\mathbf{T}$ is canalyzing in variable y_c with canalyzing value s and canalyzed value $v = 1$, if and only if $\mathbf{T}$ includes a TV $\mathbf{t}^c$ defined as $\mathbf{t}^c = [t_1^c, \ldots, t_n^c]$ with $t_c^c = s$ and $t_i^c = -$ for all $i \in \{1, \ldots, n\} \backslash c$.*

Remark 1. To be included in $\mathbf{T}$, the TV $\mathbf{t}^c$ must not be a line of $\mathbf{T}$, but all BVs in $\mathbf{t}^c$ must appear in $\mathbf{T}$, i.e., DIF($\mathbf{t}^c$,$\mathbf{T}$) = [ ]. The empty TVL corresponds to a Boolean vector with only zeros.

Proof. If $\mathbf{T}$ is canalyzing with $v = 1$ its complement $\bar{\mathbf{T}}$ is canalyzing with $v = 0$, see Lemma 2. According to Lemma 1 the complement $\bar{\mathbf{T}}$ is orthogonal to all TVs in $\mathbf{T}$. Thus there has to be a (0,1)-combination for any pair of rows out of $\mathbf{T}$ and $\bar{\mathbf{T}}$. As proposed $\mathbf{T}$ has to include $\mathbf{t}^c$, where are only '−'s in row j except of in the c-th column. To be orthogonal to $\mathbf{t}^c$ in every line of the complemented $\bar{\mathbf{T}}$ in the cth-column there has to be the element $\bar{s}$. Thus $\bar{\mathbf{T}}$ is canalyzing with $v = 0$ according to Lemma 3.

Example 1 (continued). The BF (1) from the running example is canalyzing with canaylzing variable y_1 as well as y_2, both with canalyzing value '0' and canalyzed value of '1': if y_1 or y_2, respectively, takes the value '0', than the result of the Boolean function is '1', independently of the other variable. This is also obvious from the OTVL representations in (6), which fall in the class of OTVLs described in Lemma 4.

4.3 Zhegalkin Identification with Canalyzing Constraints

In [2,4–6] it is shown how to express canalyzing functions as Zhegalkin polynomials and integrate those constraints in the Zhegalkin identification. Here it is shown how to restrict the Branch-and-Cut algorithm in Sect. 3 to canalizing constraints. In addition to its good biological properties another worthwhile advantage of canalyzing functions is their reduced number compared to all Boolean functions, see Table 1. There the number of canalyzing Boolean functions for n variables is compared all existing Boolean functions. A significant decrease of the number of canalyzing functions compared to all Boolean ones is obvious. The adaption introduced here of the identification algorithm takes advantage of that and can considerably reduces the calculation time thereby.

To restrict the Branch-and-Cut algorithm from [15] to canalyzing functions, only few adaptions are necessary. First instead of initializing the search tree with the empty OTVL as before, it is to initialize with the $2n$ TVs of n variables, which are canalyzing with $v = 1$ (Fig. 2).

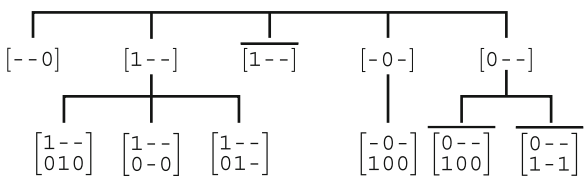


Fig. 2. Search tree for Boolean identification restricted to canalyzing functions with $n = 3$ and row number restricted to two, [3].

Example 2. For 3 variables, due to Lemma 4 all TVs, which are canalyzing with $v = 1$ are given as

$$[1\text{--}]\,, \quad [\text{-}1\text{-}]\,, \quad [\text{--}1]\,, \quad [0\text{--}]\,, \quad [\text{-}0\text{-}]\,, \quad [\text{--}0]\,,$$

where the canalyzing variable of the two TVs in the first columns is the first variable with the canalyzing value 1 and 0, e.g., for the second and third variable.

Due to Lemma 4 any orthogonal TVs can be added to these root-nodes, without loosing the canalyzing property. Furthermore each existing canalyzing function with $v = 1$ (with respect to the maximum line constraint) is in the search space, because by initialization all existing combinations of canalyzing variable and value are covered, and can thus be identified.

To cover also the canalyzing functions with $v = 0$ as additional roots those $2n$ TVs, which are canalyzing with $v = 1$, are taken again, but subtracted from the TV only consisting of '$-$'s, describing the whole Boolean space. Note that the subtraction operation for Zhegalkin polynomials is equivalent to the Difference operation for the corresponding OTVLs. Subtracting a TV of the whole Boolean space is equivalent to building the complement, thus due to Lemma 2 the resulting OTVL is canalyzing with $v = 0$. If one of these root-nodes with $v = 0$ should be branched, then instead of adding all orthogonal TVs, all orthogonal TVs are subtracted. Hereby the canalyzing property with $v = 0$ is preserved. Note that for checking if a TV is orthogonal, it is more efficient to check if it is orthogonal to all TV's that are substracted, then from the difference itself. To distinguish between the OTVLs canalyzing with $v = 1$ and $v = 0$, v is added as further variable to each node. In the branching step, if for the branching node we have $v = 1$, orthogonal TVs have to be added, otherwise subtracted. For the cutting step, the cutting condition also depends on v as follows

$$\text{cut node } j \quad \begin{cases} \text{with } v = 0 & \text{if } \exists\, t \in \{1, ..., T\} : \; d_j(t) > J^+ \\ \text{with } v = 1 & \text{if } \exists\, t \in \{1, ..., T\} : \; d_j(t) > -J^+. \end{cases}$$

5 Nested Canalyzing Functions

NCFs are a natural specialization of CFs, [8], considering the case, when a CF does not get the canalyzing input. If in that case, the restricted function is again canalyzing in another variable etc., it is said to be nested canalyzing, [19].

Definition 4 ([8]). *Let f be a BF in n variables and σ a permutation order on $1, \ldots, n$. The function f is nested canalyzing in the variable order $x_{\sigma(1)}, \ldots, x_{\sigma(n)}$ with canalyzing value vector $\mathbf{s} = [s_1, \ldots, s_n] \in \mathbb{B}^n$ and canalyzed value vector $\mathbf{v} = [v_1, \ldots, v_n] \in \mathbb{B}^n$, if*

$$f(x_1,\ldots,x_n)=\begin{cases} v_1 \;\; if\; x_{\sigma(1)}=s_1\,, \\ v_2 \;\; if\; x_{\sigma(1)}=\bar{s}_1 \wedge x_{\sigma(2)}=s_2\,, \\ v_3 \;\; if\; x_{\sigma(1)}=\bar{s}_1 \wedge x_{\sigma(2)}=\bar{s}_2 \wedge x_{\sigma(3)}=s_3\,, \\ \quad\vdots \qquad\quad \vdots \\ v_n \;\; if\; x_{\sigma(1)}=\bar{s}_1 \wedge \cdots \wedge x_{\sigma(n-1)}=\bar{s}_{n-1} \wedge x_{\sigma(n)}=s_n\,, \\ \bar{v}_n \;\; if\; x_{\sigma(1)}=\bar{s}_1 \wedge \cdots \wedge x_{\sigma(n)}=\bar{s}_n\,. \end{cases} \tag{12}$$

If f is nested canalyzing with canalyzing value vector $\mathbf{s}=[s_1,\ldots,s_n]$ and canalyzed value vector $\mathbf{v}=[v_1,\ldots,v_n]$, it is also nested canalyzing with canalyzing value vector $\mathbf{s}=[s_1,\ldots,\bar{s}_n]$ and canalyzed value vector $\mathbf{v}=[v_1,\ldots,\bar{v}_n]$.

Analogously to Lemma 2 for CFs, the following can be stated and proven for NCFs.

Lemma 5. *Given a BF f of n variables, which is nested canalyzing in variable order $x_{\sigma(1)},\ldots,x_{\sigma(n)}$ with canalyzing value vector $\mathbf{s}=[s_1,\ldots,s_n]$ and canalyzed value vector $\mathbf{v}=[v_1,\ldots,v_n]$, then the complement function $\bar{f}$ is nested canalyzing in variable order $x_{\sigma(1)},\ldots,x_{\sigma(n)}$ with $\mathbf{s}=[s_1,\ldots,s_n]$ and $\bar{\mathbf{v}}=[\bar{v}_1,\ldots,\bar{v}_n]$.*

The application of this Lemma is shown for a small example with $n=3$ in Table 4.

Table 4. Properties of a nested canalyzing OTVL and its complement.

	$\mathbf{T_1}$	$\mathbf{T_2}=\bar{\mathbf{T}}_\mathbf{1}$
OTVL	$\begin{bmatrix}1\text{--}\\000\end{bmatrix}$	$\begin{bmatrix}001\\01\text{-}\end{bmatrix}$
permutation order σ	$\{x_1,x_2,x_3\}$	$\{x_1,x_2,x_3\}$
canalyzing value vector $\mathbf{s}$	(111)	(111)
canalyzed value vector $\mathbf{v}$	(100)	(011)
graphical representation		

According to [8] the number $N_{\mathrm{nc}}(n)$ of all NCFs for a given n is calculated iteratively by

$$N_{\mathrm{nc}}(n)=2\,E(n) \quad \text{with}$$

$$E(1)=1,\quad E(2)=4,\quad E(n)=2^n+\sum_{r=2}^{n-1}\binom{n}{r-1}2^{r-1}E(n-r+1)\quad \forall n\geq 3. \tag{13}$$

Due to the iterative Definition 4 testing an OTVL of being nested canalyzing is an iterative application of canalyzing tests. If a variable has been found in

which the OTVL is canalyzing, possibly a difference operation is necessary if $v = 1$. Then the OTVL reduced by the respective column has to be tested again, until no column is left or the reduced OTVL is not canalyzing. Such a test can be implemented as a depth-first search algorithm. It is costly; in worst case n canalyzing tests have to be performed.

The Branch-and-Cut identification algorithm restricted to CFs, as discussed in Sect. 4.3, can not be further restricted to NCFs. The reason therefore is that a constriction of NCFs, similar as for CFs, that preserves the nested canalyzing property when a OTV is added, is not possible for NCFs.

On the other hand, enforcing the nested canalyzing property by testing each node of the Branch-and-Cut identification algorithm restricted to CFs for being nested canalyzing is not appropriate either, since, as discussed above, this operation is costly and would increase the complexity and thus the calculation time enormously.

To avoid the testing in the algorithm, as alternative, identification by enumeration is proposed here. This is possible due to small number of NCFs compared to CFs, see Table 1. For enumeration, the generation of all existing nested canalyzing OTVLs for a given n is necessary. A respective algorithm is given in Algorithm 1. As a prerequisite the following results are needed.

Proposition 4. *Let f be a BF of n variables and let $\mathbf{b} \in \mathbb{B}^{2n}$ be the corresponding truth vector. If f defines a nested canalyzing function, then $\mathbf{b}$ has an odd number of ones, N_1, and an odd number of zeros, N_0, respectively.*

Proof. This can be proven by Definition 4. By fixing one variable x_{σ_1} to s_1 as in (12), one half space of the Boolean space $\mathbb{B}^{2^n}$ is set to v_1, i.e., $2^{(n-1)}$ elements in $\mathbf{b}$ are set to v_1, which is an even number if $n > 1$. According to the definition, for the remaining half space, half of the elements are set to v_2 by fixing x_{σ_2} to s_2, etc.

In general, a Boolean space $\mathbb{B}^{2^{(n-i+1)}}$ consists of an even number of elements as well as the respective Boolean half space $\mathbb{B}^{2^{(n-i)}}$, if $i < n$. Therefore in (12) by iteratively fixing the variable x_{σ_i} to s_i, in iteration i an even number of elements is set to v_i for $i \in \{1, \ldots, n-1\}$. For $i = n$ only one element of the two remaining elements of $\mathbb{B}$ is set to v_n. So the total number of ones and zeros respectively is uneven. □

Proposition 5. *Let $\mathbf{T} = [t_{11} \cdots t_{1n}]$ be an OTVL with one line and without any don't cares, then the corresponding BF $f(x_1, \ldots, x_n)$ is nested canalyzing in all possible permutation orders σ with canalyzing value vector $\mathbf{s} = [\bar{t}_{11}, \ldots, \bar{t}_{1n}]$ and canalyzed value vector $\mathbf{v} = [0, \ldots, 0]$.*

Proof. In Proposition 4 it has been stated that a BF can only be nested canalyzing having an uneven number of ones and zeros respectively. Therefore, according to Proposition 2 for OTVLs with one line only those without any *don't cares* can be nested canalyzing. This does not allow the conclusion, that they have to be nested canalyzing, but it can be shown according to Theorem 3, where it is stated, that an OTVL is canalyzing in x_i with $s = \bar{t}_{1i}$ and $v = 0$, if $t_{1i} = t_{2i} = \cdots = t_{ki}$,

where k is the number of lines. In an OTVL with only one line this is valid for all n columns if there is no *don't care*. Therefore, an OTVL with one line is canalyzing in every possible variable x_i with $s = \bar{t}_{1i}$ and $v = 0$. Reducing this OTVL by column i does not change this characteristics, so the same is valid for the reduced OTVL as well. Thus the OTVL is nested canalyzing. □

For $n = 1$ there are two NCFs, see (13). Due to Proposition 4 the corresponding BFs have to have an odd number of ones and zeros. Therefore, with Proposition 2, it can be concluded that [ ] and [-] are not NCFs for $n = 1$, but $\mathbf{T}_{1,1} = [1]$ and its complement $\mathbf{T}_{1,2} = \bar{\mathbf{T}} = [0]$, according to Lemma 5. For $n = 2$, (13) states that there are 8 NCFs. For $n = 2$ there are four OTVLs with one line and without any *don't cares*. Due to Proposition 5 these are nested canalyzing and, according to Lemma 5, their complements as well. Thus, all NCFs for $n = 2$ are given by

$$\begin{aligned} &\mathbf{T}_{2,1} = [00]\ , \quad \mathbf{T}_{2,2} = [10]\ , \quad \mathbf{T}_{2,3} = [01]\ , \quad \mathbf{T}_{2,4} = [11]\ , \\ &\mathbf{T}_{2,5} = \bar{\mathbf{T}}_{2,1}, \quad \mathbf{T}_{2,6} = \bar{\mathbf{T}}_{2,2}\ , \quad \mathbf{T}_{2,7} = \bar{\mathbf{T}}_{2,3}\ , \quad \mathbf{T}_{2,8} = \bar{\mathbf{T}}_{2,4}\ . \end{aligned} \tag{14}$$

The set of nested canalyzing OTVLs for $n = 2$, $F_{\mathrm{nc}}(2)$, is the prerequisite to build the sets of all nested canalyzing OTVLs with $n > 2$ variables recursively, as it is also the case for the calculation of the respective set sizes (13).

Theorem 1. *Given are the sets of all nested canalyzing OTVLs with* $2, ..., n-1$ *variables as* $F_{\mathrm{nc}}(2), ..., F_{\mathrm{nc}}(n-1)$, *which are composed as* $F_{\mathrm{nc}}(i) = \{F^p_{\mathrm{nc}}(i), \bar{F}^p_{\mathrm{nc}}(i)\}$. *Here* $\bar{F}^p_{\mathrm{nc}}(i)$ *contains the complement of all OTVLs in* $F^p_{\mathrm{nc}}(i)$. *Then the set of nested canalyzing OTVLs for n variables are derived according to Algorithm 1 .*

Proof. It is to prove (i) that the number of all generated OTVLs by Algorithm 1 is $N_{\mathrm{nc}}(n)$ in (13), (ii) that the generated OTVLs are all NCFs and (iii) that the generated OTVLs all represent different BFs.

To show (i) note that both in line 2 and 9 of Algorithm 1 an OTVL is added to the set $F^p_{\mathrm{nc}}(i)$. Line 2 is called 2^n times and line 9 is called $(\sum_{r=2}^{n-1} \binom{n}{r-1} 2^{r-1} E(n-r+1))$ times, what adds up to $E(n)$. In line 14, the number of OTVLs is doubled by considering the complements. This leads to $N_{\mathrm{nc}}(n)$ and proves (i).

The statements (ii) and (iii) are proven by complete induction. Obviously the given OTVLs for $n = 2$ in (14) are nested canalyzing and represent different BFs. Assume that all OTVLs for N variables generated with Algorithm 1 are nested canalyzing and represent different BFs. This implies that this holds for $<N$ variables, since the OTVLs for N are generated based upon those. To get the OTVLs of $(N + 1)$ variables the algorithm adds $(N - r + 1)$ columns to those OTVLs of $r \leq N$, that consist only of '1's or '0's, so that the resulting OTVL is canalyzing in the respective variable (see Theorem 3). Thus all OTVL for $(N+1)$ are NCFs, what proofs (ii). Assume all $\bar{F}^p_{\mathrm{nc}}(i_1)$ are different BFs. Due to construction none of those has a columns with only '1's or '0's. Thus when $N-i_1+1$ of those columns are added, to generate OTVLs for $N+1$ variables from the elements in $\bar{F}^p_{\mathrm{nc}}(i_1)$, the 2^{N-n_1+1} different valid assignments and different placements result in different Boolean functions with different canalyzing value

Algorithm 1. Generate all nested canalyzing OTVLs for n, given $F_{\mathrm{nc}}(2), \ldots, F_{\mathrm{nc}}(n-1)$.

1: $F^p_{\mathrm{nc}}(n) = \{\}$: define empty set
Generate NCFs for n variables and 1 line
Iterate over all 2^n variations to fill n places with 0's and 1's
2: **for** $i = 1$ **to** 2^n consider $\mathbf{T}_{n,i}$ as the i's of those variations, set $F^p_{\mathrm{nc}}(n) = \{F^p_{\mathrm{nc}}(n), \mathbf{T}_{n,i}\}$
3: **end for**
Generate NCFs for n variables and >1 lines
4: $i = 2^n + 1$
Iterate over the NCFs with $n-(r-1)$ variables with r between 2 and $n-1$
5: **for** $r = 2$ **to** $n-1$
6: *Iterate over all NCFs in $\bar{F}^p_{\mathrm{nc}}(n-(r-1))$ with $n-(r-1)$ variables*
7: **for** $t = E(n-(r-1))+1$ **to** $2E(n-(r-1))$ consider $\mathbf{T}_{n-(r-1),t}$,
Iterate over all combinations to place the $n-(r-1)$ columns of $\mathbf{T}_{n-(r-1),t}$ in the n columns of $\mathbf{T}_{n,i}$
8: **for** $l = 1$ **to** $\binom{n}{r-1}$ place $\mathbf{T}_{n-(r-1),t}$ at the l's place combination of $\mathbf{T}_{n,i}$,
Iterate over all 2^{r-1} combinations to fill remaining $r-1$ columns with equal rows
9: **for** $k = 1$ **to** 2^{r-1} fill remaining columns of $\mathbf{T}_{n,i}$ with the k's combination,
10: set $F^p_{\mathrm{nc}}(n) = \{F^p_{\mathrm{nc}}(n), \mathbf{T}_{n,i}\}$, $i = i+1$
11: **end for**
12: **end for**
13: **end for**
14: **end for**
15: $F_{\mathrm{nc}}(n) = \{F^p_{\mathrm{nc}}(n), \bar{F}^p_{\mathrm{nc}}(n)\}$

vectors $\mathbf{s}$ and permutation orders σ. Given $\bar{F}^p_{\mathrm{nc}}(i_2)$ with $i_1 \neq i_2$, all OTVLs with $N+1$ generated by those have $N - i_2 + 1$ columns with only '1's or '0's, thus represent different BFs than those based on $\bar{F}^p_{\mathrm{nc}}(i_1)$ with $N - i_1 + 1$ such columns. □

Example 3. We will show exemplarily how to generate all nested canalyzing OTVLs for $n = 3$ with Algorithm 1. In line 2 to 3 of the algorithm all OTLVs with one line are generated. For $n = 3$, there are 8 of them given as

$$\mathbf{T}_{3,1} = [000]\ , \quad \mathbf{T}_{3,2} = [100]\ , \quad \mathbf{T}_{3,3} = [010]\ , \quad \mathbf{T}_{3,4} = [001]\ ,$$
$$\mathbf{T}_{3,5} = [011]\ , \quad \mathbf{T}_{3,6} = [110]\ , \quad \mathbf{T}_{3,7} = [101]\ , \quad \mathbf{T}_{3,8} = [111]\ .$$

The for-loop in line 5 is only called once, since $n - 1 = 2$. The for-loop in line 7 is called four times for the NCFs in $\bar{F}^p_{\mathrm{nc}}(2)$, which are given in (14) as $\mathbf{T}_{2,5}$, $\mathbf{T}_{2,6}$, $\mathbf{T}_{2,7}$ and $\mathbf{T}_{2,8}$. Let's consider only the case $\mathbf{T}_{2,5}$ as an example.

To get an OTVL for $n = 3$ an additional column has to be added, as first, second or third column. The iteration over these possibilities is realized in line 8. There are two choices for the column to be added, either the one of only '1's or of only '0's. The iteration over the different possibilities is performed in line 9. All NCFs with $n = 3$ resulting from $\mathbf{T}_{2,5}$ are summarized in Table 5.

Table 5. All nested canalyzing OTVLs for $n = 3$ built by $\mathbf{T}_{2,5}$ from (14).

	Adding the new column in the		
	1. column	2. column	3. column
Choosing 1	$\begin{bmatrix}11-\\101\end{bmatrix}$	$\begin{bmatrix}11-\\011\end{bmatrix}$	$\begin{bmatrix}1-1\\011\end{bmatrix}$
Choosing 0	$\begin{bmatrix}01-\\001\end{bmatrix}$	$\begin{bmatrix}10-\\001\end{bmatrix}$	$\begin{bmatrix}1-0\\010\end{bmatrix}$

This can be done analogously for $\mathbf{T}_{2,6}$, $\mathbf{T}_{2,7}$ and $\mathbf{T}_{2,8}$ from (14).

6 Application of the Constrained Identification Algorithm

The presented identification algorithm is applied to gene expression data also used in [3,5,15]. The considered gene expression data are measurements of mRNA extracted from mouse liver cells using microarray technology (GeneChip Human Exon 1.0 ST Array). The measurements were repeated four times (T = 3) after 2, 4, 12 and 24 hours [18]. In total the expression levels of 21799 genes could be detected. Three different mRNA samples were tested, one treated with the contaminant Benzo(a)pyrene (BaP) with a concentration of 5μM, one with a lower one of 50nM and one control sample without treatment. The samples will be called **T5μM**, **T50nM** and **Control** in the following. This contaminant BaP is found in cigarette smoke and automobile exhaust and is connected to deadly diseases such as cancer. Geneticists assume that the contamination of cells with BaP with the high concentration of 5μM leads to the cellular process apoptosis, programmed cell death, but not the contamination with the low concentration. Therefore the present gene data is analyzed with regard to apoptosis.

The apoptotic pathway for mice can be found in the KEGG database, [10], hosted by Kanehisa Laboratory. From all detected genes, 78 are, due to the database, known to be involved in the apoptotic pathways. These are extracted and considered in the following. The database gives for each gene a set of genes where it may depend on. This knowledge is taken into account for a first identification, where these sets are taken as possible clusters for the identification of the respective gene. Thereby possible solutions of clusters are a priori reduced. The identification without constraints as given in [15] is applied, the adapted one with canalyzing constraints as presented in Sect. 4.3 and the identification by enumeration restricted to NCFs. The maximum number of rows of the resulting OTVLs is restricted to two using the Branch-and-Cut algorithm. For the identification for each gene a model for connectivity degree two up to the set size given in the database is identified with canalyzing constraints. For the identification without constraints the maximal connectivity degree for each gene is restricted to 5 due to exploding calculation time. For the same reason, the maximal connectivity degree of 5 is also set for the identification with nested canalyzing constraints. Remind, that while the constraint to canalyzing functions reduces the search

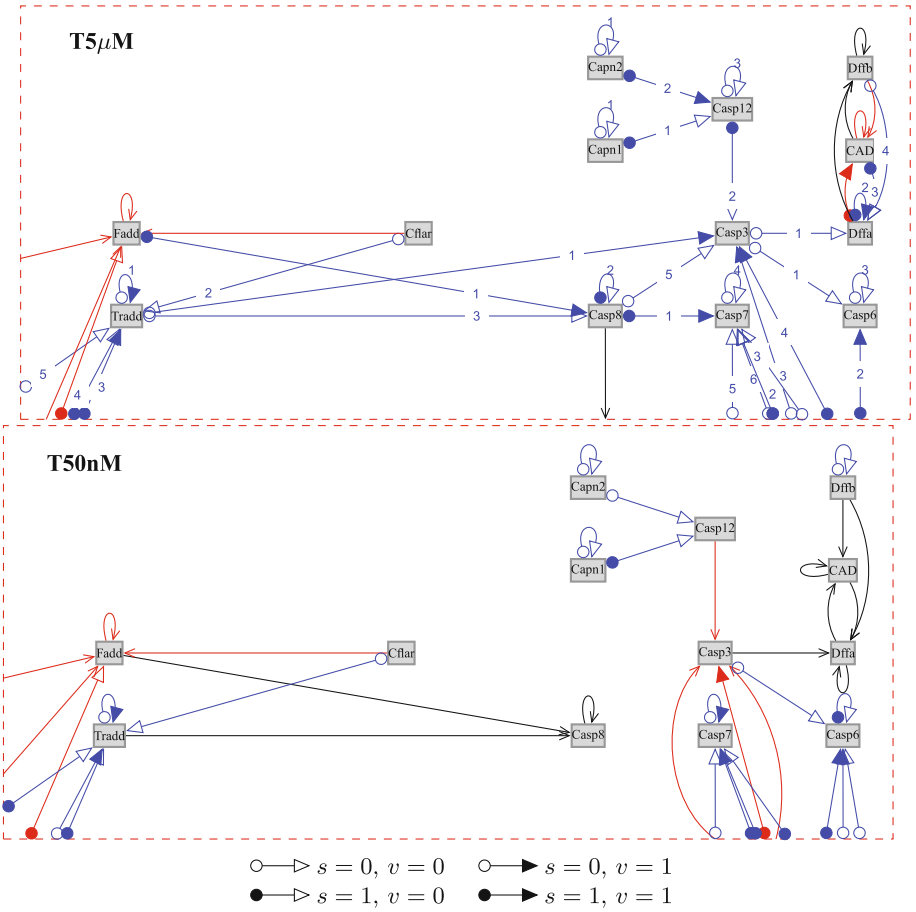


Fig. 3. Identified extrinsic pathway for **T5**μ**M** and **T50nM** with given clustering constraints, (canalyzing functions in red canalyzing functions, with no constraints in black, that with minimum error is shown).

space and thus the calculation time of the identification, for nested constraints this is not the case and enumeration is used.

A cutout of the identified network is shown in Fig. 3 for both samples with contamination. In general the apoptotic pathways consists of the extrinsic pathway and the intrinsic one. Here the extrinsic one is shown in detail. The expectation, that the concentration of **T5**μ**M** leads to apoptosis, while that of **T50nM** does not, is affirmed here. According to the database the extrinsic pathway is triggered by engagements at the death ligands, which activate *caspase-8*. That induces a signaling cascade, resulting in an activation of *caspase-3*, what leads to cell death. This can be seen for **T5**μ**M**, where *caspase-8* is activated leading to and activation of *caspase-3*. The red arcs with circled tail and triangular head,

denote the canalyzing genes, thus the major influencing one. For nested canalyzing functions the numbers in the blue arcs determine the canalyzing order, thus the major influencing ones have a small number. If the tail is colored, its canalyzing value is one, if the head is colored, the canalyzing value is one, and zero otherwise. Thus, for the network of **T5μM** this means that the activation of *Fadd* activates *caspase-8* in any case, what initiates *caspase-7* independently from any other genes. A well known result in apoptosis is confirmed hereby. Correspondingly, a deactivation of *Tradd* will definitely lead to an activation of *caspase-3* and thus to apoptosis. Comparing the results to the low concentration sample **T50nM**, here, none of the interconnection, responsible for apoptosis are found. This indicates that the low concentration of BaP does not lead to apoptosis as expected.

The identification is repeated, without considering the dependency sets given by the database, but testing all possible clusters with connectivity degree from two to four. Note that thus for one gene $\binom{78}{4} + \binom{78}{3} + \binom{78}{2} = 1505504$ different clusters have to be checked. The identification with canalyzing constraints is performed with maximum number of rows of the OTVL restricted to two and the identification with nested canalyzing constraints using enumeration. The algorithm without constraints is not tractable anymore. The average errors over the models for all genes are given in Table 6 for both cases, the identification considering the dependency sets and the general one with all possible clusters and for high and low concentration. For each gene the best model identified has been considered. Two trends are recognizable in the results. First, when all possible clusters are considered, smaller errors are obtained. This is clear since the search space is increased. Second, the average errors for **T50mM** are slightly larger. This also let suspect, that the high concentration rather lead to apoptosis than the low one. Here only the genes involved in apoptosis are considered, but if other processes are executed, other genes may be involved. In general it can be said that all models fit well, since biologists talk about good approximations if an error $< 10^{-3}$ is achieved. Remark that for the identification the maximum number of lines of the identified OTVLs was restricted to two, which is necessary to reduce the solution space and make the problem tractable. This seems to be very small. To make a statement if this is enough, we have to disregard the models achieved with nested canalyzing constraints, since here due to identification by enumeration the rank can be possibly larger. The respective average error is reported in brackets in Table 6. The errors

Table 6. Root mean square errors of the identified models. For each gene the best model identified is considered. For the errors in brackets the models identified with nested canalyzing constraints are disregarded.

	Clusters constrained by biologists	All possible clusters
T50nM	$9.95 \cdot 10^{-4}$ $(1.2 \cdot 10^{-3})$	$1.24 \cdot 10^{-4}$ $(1.43 \cdot 10^{-4})$
T5μM	$9.99 \cdot 10^{-4}$ $(1.2 \cdot 10^{-3})$	$3.62 \cdot 10^{-5}$ $(6.19 \cdot 10^{-5})$

only slightly increase, which suggests that the low rank might be enough for a model of appropriate quality.

The better fit of the models achieved for the sample **T5**μ**M** are confirmed when the continuous gene expression level dynamics are analyzed. Therefore, the measurements and the prediction using the identified model are compared. The prediction of gene l, initialized with the measured values $\tilde{\mathbf{y}}(0)$, is determined by

$$y_l(t+1) = p_{\mathbf{T}_l}(\mathbf{y}(t)) \quad \text{with} \quad \mathbf{y}(0) = \tilde{\mathbf{y}}(0).$$

The dynamics of two genes for **T5**μ**M** and **T50nM** are shown in Fig. 4. Here with *caspase-6* and *caspase-8*, two genes right in the center of the extrinsic pathway are depicted. The predictions for **T5**μ**M** fit very well, what is not astonishing since errors of $5.3 \cdot 10^{-5}$ and $1.4 \cdot 10^{-8}$ are achieved. For the sample **T50nM** the error of the identified model is with $1.8 \cdot 10^{-4}$ and $3.2 \cdot 10^{-4}$ also small but compared to **T5**μ**M** up to 10^3-times worse, which is confirmed by the prediction. This also supports the conclusion, that other processes then apoptosis with other genes involved might occur for that sample.

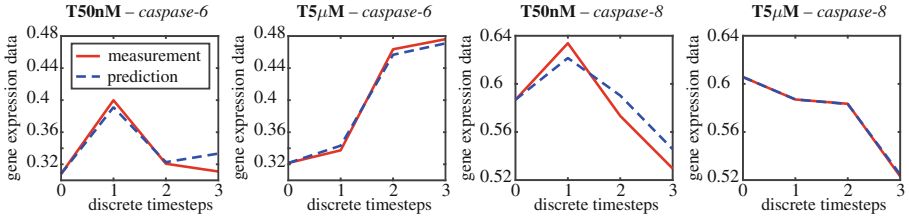


Fig. 4. Measured vs. predicted gene expression level dynamics for **T50nM** and **T5**μ**M**.

To analyze the long term behavior, predicted for 20 time steps are simulated. The simulation of *caspase-6* is exemplarily shown in Fig. 5. One simulation is performed with the actually measured initial condition, shown in red. In addition, simulations with perturbed initial condition are shown in blue. While for **T50nM** and **T5**μ**M** the performance of the disturbed prediction is not only stable but also very robust, this is not the case for **Control**, where the disturbed trajectories show a chaotic behavior. In **T50nM** all trajectories converge to the measured initial value, while in **T5**μ**M** all trajectories also converge, but the prediction of the measured initial value performs a step. This may indicate that some reaction is taken place.

It has been stated before that for some identification methods the connectivity degree had to be limited for calculation time reasons. To analyze the calculation time in further depth, the average calculation times for one cluster are shown in Table 7 for the different connectivity degrees and identification methods. In general it can be seen, that the calculation time rises with higher connectivity degree. Only for the very small ones this is not that clear to see. It is assumed, that here the calculation time is mainly effected by initialization steps

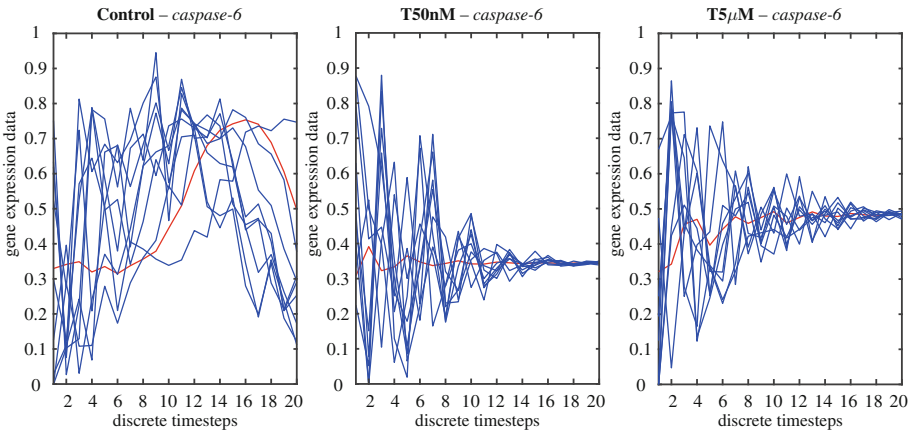


Fig. 5. Long term behavior of *caspase-6* in simulation.

Table 7. Comparison of the mean calculation time per cluster for the identification with the preselected set of possible clusters proposed by the biologists and the identification over all possible clusters. Empty items have not been calculated due to calculation time issues.

Connectivity degree	Canalyzing		ncf		No constraints	
	Preselected clusters	All clusters	Preselected clusters	All clusters	Preselected clusters	All clusters
2	0.0105 s	0.0098 s	0.0195 s	0.0003 s	0.0195 s	–
3	0.0022 s	0.0030 s	0.0015 s	0.0023 s	0.0268 s	–
4	0.0057 s	0.0161 s	0.0126 s	0.0370 s	0.7213 s	–
5	0.0222 s	–	0.2197 s	–	71.0064 s	–
6	0.5548 s	–	–	–	–	–
9	0.1870 h	–	–	–	–	–
11	19.9031 h	–	–	–	–	–

and is therefore equal for all connectivity degrees. As a second observation it can be concluded, that the identification with canalyzing constraints is considerably faster than without constraints. This is clear since the search space has been considerably reduced. The calculation time with nested canalyzing constraints is somewhere in between. Although the ratio of nested canalyzing function compared to all, and also to all canalyzing functions, is very small as shown in Table 1, the Branch-and-Cut algorithm exploiting rank restriction is not possible, but enumeration has to be performed, as described in Sect. 5. Therefore, the calculation time is in average larger than with canalyzing constraints at least for connectivity degrees larger than 2.

Remind that for the identification without constraints with the preselected set of possible clusters proposed by the biologists the maximal connectivity degree for each gene is restricted to 5, although for some genes the preselected set is larger. But already for a connectivity of 5 the average calculation time for one possible cluster is with 71 s more than a minute. And if a gene may depend on 11 genes, according to suggestions of the biologists, with a connectivity degree of 5 this results in $\binom{11}{5} = 462$ possible clusters, and thus in more than 546 min for only one gene. In comparison, with canalyzing constraints one cluster takes 0.022 s for a connectivity degree of 5 leading to approximately 10 s in total. For a connectivity degree of 11, the maximum one found in the database, the identification with canalyzing constraints takes $28.66 \cdot 10^3$ s. Note that when all possible clusters are considered, there are 1426425 possible clusters. With an average calculation time of 0.0161 s this results in 6.4 h only for one gene for canalyzing constraints. For nested canalyzing constraints the time is $\approx 2.5\times$ as long for one gene.

7 Conclusions

The paper presents how to express canalyzing and nested canalyzing functions in terms of OTVLs. Based on the definition of canalyzing OTVLs, it is shown how to restrict the solution space of the Branch-and-Cut algorithm for Boolean identification in [15] to canalyzing functions by simple adaptions mainly in the initialization step. Thereby the restriction to a maximum number of lines, that as a core of the algorithm leads efficiently to a suboptimal solution, does not need to be given up. The advantage of the restriction to canalyzing function is twofold, first from the biological point of view, since canalyzing functions are known to describe gene networks better than other functions, and second from the computational point of view. By the adaption of the Zhegalkin identification algorithm presented in this paper, the search space is enormously reduced by the canalyzing constraints, what leads to manageable computation times even for larger data.

Nested canalyzing functions are a subclass of canalyzing function, which have drawn the interest of biologists due to their favorable properties for gene identification. However due to their iterative definition, the line restriction exploited in the Branch-and-Cut algorithm in [15] can not be applied. Instead, identification by enumeration is proposed here. Therefore, an algorithm for the generation of all nested canalyzing OTVLs is introduced.

The presented algorithms have been applied to experimental gene data. The Boolean identification restricted to canalyzing and nested canalyzing function result in well fitting models with desirable properties like stability and robustness, while restricting the calculation time significantly compared to Boolean identification without constraints. The reduction in calculation time makes the considered large scale problem with 78 genes tractable also for higher connectivity degrees tractable and allows to incorporate the clustering step in the identification. Furthermore assumptions of the biologists regarding the network

structure and the importance of specific genes for the specific process apopotosis could be approved by the algorithm presented here.

Acknowledgements. The authors would like to thank Saskia Trump and Sabine Attinger from Helmholtz Center for Environmental Research Leipzig for access to microarray data.

References

1. Bochmann, D., Steinbach, B.: Logikentwurf mit XBOOLE. Verlag Technik (1991)
2. Breindl, C., Chaves, M., Allgöwer, F.: A linear reformulation of Boolean optimization problems and structure identification of gene regulation networks. In: Proceedings of 52nd IEEE Conference on Decision Control, pp. 733–738 (2013)
3. Eichler, A., Lichtenberg, G.: Parameter identification of canalyzing Boolean functions with ternary vectors for gene networks. In: Proceedings of 6th International Conference on Simulation and Modelling Methodologies, Technologies and Applications (Simultech), vol. 1, pp. 110–118 (2016)
4. Faisal, S.: Discrete-Time Modelling of Gene Networks by Zhegalkin Polynomials. Dr. Hut Verlag, Ingenieurwissenschaften (2008)
5. Faisal, S., Lichtenberg, G., Trump, S., Attinger, S.: Structural properties of continuous representations of Boolean functions for gene network modelling. Automatica **46**(12), 2047–2052 (2010)
6. Faisal, S., Lichtenberg, G., Werner, H.: A polynomial approach to structural gene dynamics modelling. In: Proceedings of 16th IFAC World Congress, p. 2119 (2005)
7. Faisal, S., Lichtenberg, G., Werner, H.: Canalizing Zhegalkin polynomials as models for gene expression time series data. In: Proceedings of 1st International Congress on Engineering and Intelligence Systems (2006)
8. Jarrah, A.S., Raposa, B., Laubenbacher, R.: Nested canalyzing, unate cascade, and polynomial functions. Physica D Nonlinear Phenom. **233**(2), 167–174 (2007)
9. Just, W., Shmulevich, I., Konvalina, J.: The number and probability of canalizing functions. Physica D Nonlinear Phenom. **197**(3–4), 211–221 (2004)
10. Kanehisa, M., Goto, S.: KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Res. **28**(1), 27–30 (2000)
11. Kauffman, S.: The Origins of Order: Self Organization and Selection in Evolution. Oxford University Press, New York (1993)
12. Kauffman, S.A., Petersen, C., Samuelsson, B., Troein, C.: Random Boolean network models and the yeast transcriptional network. PNAS **100**(25), 14796–14799 (2003)
13. Kauffman, S.A., Peterson, C., Samuelsson, B., Troein, C.: Genetic networks with canalyzing Boolean rules are always stable. PNAS **101**(49), 17102–17107 (2004)
14. Kolda, T., Bader, B.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
15. Lichtenberg, G., Eichler, A.: Multilinear algebraic Boolean modelling with tensor decompositions techniques. In: Proceedings of 18th IFAC World Congress, pp. 5603–5608 (2011)
16. Lichtenberg, G., Faisal, S., Werner, H.: Ein Ansatz zur dynamischen Modellierung der Genexpression mit Shegalkin-Polynomen. Automatisierungstechnik **53**, 589–596 (2005)

17. Lin, P., Khatri, S.: Logic Synthesis for Genetic Diseases. Modeling Disease Behavior Using Boolean Networks. Springer, New York (2013)
18. Michaelson, J.J., Trump, S., Rudzok, S., Gräbsch, C., Madureira, D.J., Dautel, F., Mai, J., Attinger, S., Schirmer, K., von Bergen, M., Lehmann, I., Beyer, A.: Transcriptional signatures of regulatory and toxic responses to benzo-[a]-pyrene exposure. BMC Genom. **12**(1), 502 (2011). doi:10.1186/1471-2164-12-502
19. Schober, S., Mir, K., Bossert, M.: Reconstruction of Boolean genetic regulatory networks consisting of canalyzing or low sensitivity functions. In: Proceeding of International ITG Conference on Source Channel Coding (SCC) (2010)
20. Veliz-Cuba, A., Jarrah, A.S., Laubenbacher, R.: Polynomial algebra of discrete models in systems biology. Bioinformatics **26**(13), 1637–1643 (2010)
21. Zhegalkin, I.: Arithmetics of symbolic logic. Mat. Sb. **35**(3–4), 311–377 (1928)

The Power of Surrogate-Assisted Evolutionary Computing in Searching Vaccination Strategy

Zong-De Jian, Tsan-sheng Hsu(✉), and Da-Wei Wang(✉)

Institute of Information Science Academia Sinica, Taipei 115, Taiwan
{zdjian1988,tshsu,wdw}@iis.sinica.edu.com

Abstract. We propose to use genetic algorithms to search for the best vaccination strategy for a given scenario using the output of the simulation program as fitness score. The efficacy of vaccine varies significantly. Therefore, the real challenge is to find a good strategy without a priori knowledge of the efficacy of the vaccine. We use surrogate function instead of real simulation to achieve 1000 times speedup. The average of the absolute value of errors is less than 0.5% and the rank correlation coefficient is greater than 0.93 for almost all the scenarios. The optimal solution with surrogate has fitness value very close to one using simulation. The difference is generally less than one percent. Our search results confirm the convention wisdom to vaccinate school children first. It also reveals that there is appropriate strategy which works for most scenarios. It would be interesting to build autonomous software searches through the scenario space and adaptively revise the surrogate to produce better search results.

Keywords: Vaccination strategy · Simulation for disease control · Surrogate-based genetic algorithm

1 Introduction

Agent-based stochastic simulation is an established approach for the study of infectious diseases. The flexibility to incorporate important concepts into simulation model is one of the advantage to such approach. However, it still needs a significant amount of computing resources sometimes. Epidemiologists usually have to carefully craft the scenarios to demonstrate their points. Vaccination is one of the important means to mitigate pandemic flu, thus it is vital to determine the vaccination priority with limited amount of vaccine. Instead of evaluating a few options, we formulate it as an optimization problem and use genetic algorithm to search for the best vaccination priority. The search space can contain

An earlier extended abstract of this paper appears in [1].
T.-S. Hsu—Supported in part by MOST of Taiwan Grants 104–2221-E-001-021-MY3.
D.-W. Wang—Supported in part by MOST of Taiwan Grants 105-2221-E-001-034.

M.S. Obaidat et al. (eds.), *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing 676,
https://doi.org/10.1007/978-3-319-69832-8_13

many dimensions, for example, house-hold structure is one of the important dimensions [2]. Here we focus on the dimension of vaccine efficacy.

The vaccine efficacy (VE) is a measure of relative risk (RR) that generally takes the form $VE = 1-RR$. The absolute efficacy of a vaccine compares relative risk in a vaccinated group with that in a control group [3]. Two important measures for vaccine efficacy are vaccine efficacy for susceptibility (VE_s), that is the relative risk a vaccinated individual being infected, and vaccine efficacy for infectiousness (VE_i), that is the relative risk of an individual being infected by a vaccinated one. Vaccine efficacy varies significantly, for example, Basta et al. categorized several reports of influenza vaccine trail, and estimated that the VE_s ranges from 0.08 to 0.79 [3].

With limited amount of available vaccine, the infectious disease control agency has to determine the amount of vaccine allocated to various groups. Usually the health care professionals has the highest priority and then the agency can use policy tools to distribute vaccines to different age groups. There are different objectives to choose vaccination strategies. Two objectives are studied in this paper. One is to reduce the total number of infected individuals [1], and the second is to reduce the economical impact of the epidemic [4]. We focus on the distribution of vaccine among different age groups and search for the distributions which optimize the objective functions. For a given scenario, that is the setting of our simulation module, the gene encodes the vaccine distribution among age groups and the fitness function can be one of the two objective functions. The fitness evaluation is done by running the simulation module first.

Each simulation run takes about 3 min, thus the fitness evaluation becomes the bottleneck of the optimization process. Using a faster approximation function in place of the true fitness function, in our case the simulation program, is called surrogated-assisted evolutionary computation [5]. The idea was first suggested in the mid-1980s [6]. We construct a surrogate function, which combines table lookups and linear interpolations.

For each objective function, we study 9 different vaccine efficacy settings, both VE_s and VE_i are enumerated from 0.1 to 0.9 with the increment equal to 0.4. For each setting, the genetic algorithm with simulation as well as surrogate as fitness function are applied to search for the optimal solutions. For both objectives, the top solutions for both cases point to allocate more vaccine to school-age children, which confirms the results in the literature [7]. However, we do observe that when the objective is total economic impact, when VE_s increases the amount of vaccine allocated to elementary school children decreases and young adults increases.

The fidelity of the surrogate function is studied. For both objectives, the difference between the output of surrogate function and the simulation divided by the output of simulation is less than one percent in average, the worst case is less than four percent and the average of the absolute value of error is also less than one percent.

2 Material and Method

In this paper, the simulation software that we used is developed by [8]. Below is a brief description of the simulation software which implements a stochastic discrete time agent-based model. The mock population of the model is constructed according to national demographics from Taiwan Census 2000 Data (http://eng.stat.gov.tw/). The connection between any two individuals indicates the possibility of daily and relatively close contact that could result in the successful transmission of the flu virus. An important virus-dependent parameter is the transmission probability which is denoted by p_{trans}. It is the probability that an effective contact results in an infection. A contact group is a daily close association of individuals, where every member is connected to all other members in the same group. There are eleven classes of contact groups in the model: community, neighborhood, household cluster, household, work group, high school, middle school, elementary school, daycare center, kindergarten, and playgroup [2]. The population size of Taiwan is about 22.12 million, the detail of age groups is shown in Table 1.

Table 1. The information of each age group.

Age group i	Age	Population ($\#AG_i$)	Type
1	0–5	1.72 million	Preschool children
2	6–12	2.36 million	Elementary school children
3	13–15	0.99 million	Middle school children
4	16–18	0.97 million	High school children
5	19–29	3.86 million	Young adults
6	30–64	10.28 million	Adults
7	65+	1.94 million	Elders

Each individual can belong to several contact groups simultaneously at any time. The duration of a simulation run is set at 365 days. Each day has two 12-hour periods, corresponding to daytime and nighttime. During the daytime, contact occurs in all contact groups. School-age children go to schools. There are around 7.8% school-age children do not go to school in Taiwan. They stay home in our simulation. Preschool children go to daycare center, kindergarten or playgroup. Young adults and adults go to work group. During the nighttime, contact occurs only in communities, neighborhoods, household clusters, and household. School closure policy of CDC Taiwan is also implemented. The so called 325 policy works as follow: when two symptomatic cases occurred in the same class within a 3 days interval then that class is closed for 5 days. The model parameters are similar to ones in a study by [9], with modifications to fit Taiwan situation better with the help of study outcome in contact diary study [10].

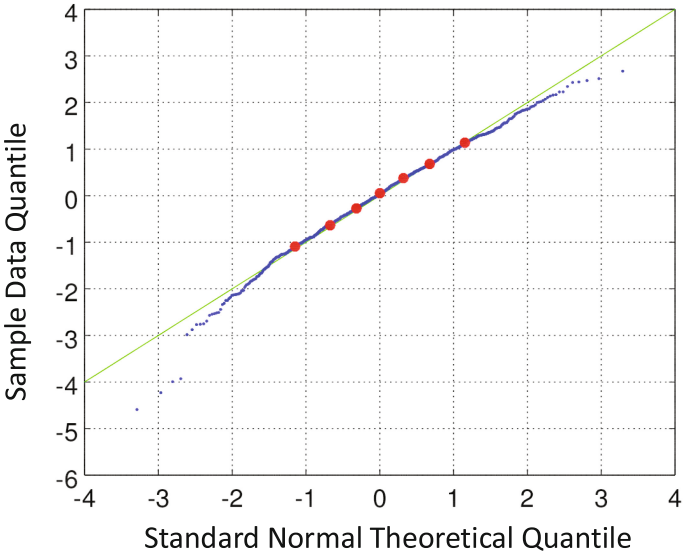


Fig. 1. The quantile-quantile (q-q) plot. [1].

In this paper, the scenario of the simulation is the following: the p_{trans} is set at 0.1, the vaccine is available 30 days after the index case occurred, total 2.5 million of doses are applied to different age groups according to the priority. Only the vaccine priority and vaccine efficacy (VE_i, VE_s) can be changed.

The simulation is a stochastic process. To assess the stochastic variability of simulation results, we carried out a thousand-run experiment for a typical baseline case, where $VE_i = VE_s = 0.5$ and each age group is allocated 500,000 doses. Similar to the finding reported in [8], the number of infected cases follows normal distribution. The quantile-quantile plot is shown in Fig. 1. The mean of the number of infected cases is 5,694,972 and standard deviation is around 10,850. These numbers serve as a reference of the stochastic variability of the simulation system, especially we take $10,850/5,694,972 \approx 0.002$ as the *coefficient of variation* of the simulation system.

There are seven age groups in our simulation and the vaccine is allocated in the unit of 10,000 doses. The total number of possible combination is $C_{250}^{250+7-1} \approx 3.69 \times 10^{11}$. An exhaustive search is not feasible. We thus use genetic algorithm with simulated annealing to search for optimal solution. The hybrid simulated annealing genetic algorithm ($HSAGA$) adds a simulated annealing component in each iteration in the genetic algorithm. The idea is to increase stochastic variability at the early stage of evolutionary step to escape local minima/maxima.

We follow the formulation of Meltzer to compute the economical impact of epidemics. In his formulation, people are divided to three age groups, zero to nineteen years old, twenty to sixty four years old and sixty five and above.

For different age group, the probability of clinic visit and hospitalization once infected are different, also the cost of treatments are different, moreover, the daily productivity lost is also different. However, the cost of vaccine and the side effect caused by vaccination are not included.

A vaccine priority is defined by $(e, \boldsymbol{X})$, where $e = (VE_i, VE_s)$ represent the vaccine efficacy and $\boldsymbol{X} = (x_1, x_2, ..., x_7)$ represent the allocation of vaccine to age groups, x_i is the amount of vaccine for age group i. We sometimes omit e when it is clear. Let p denote a vaccine priority, and we use $sim(p)$ to denote the value if the objective function reported by the simulation program with p. We use point instead of vaccination priority when there is no confusion. Let S denotes the set of points already simulated, that is for all $p \in S$ the value of $sim(p)$ is known. Let C_{basis} denote the baseline case with no vaccination, that is $C_{basis} = sim(\mathbf{0})$. We use p_i and $p_{j,k}$ to denote vectors with only nonzero dimension i and nonzero dimensions j and k respectively. We sometimes abuse the notion to use p_i and $p_{j,k}$ to denote the projection of point p to i^{th} dimension and to j^{th} and k^{th} dimensions respectively.

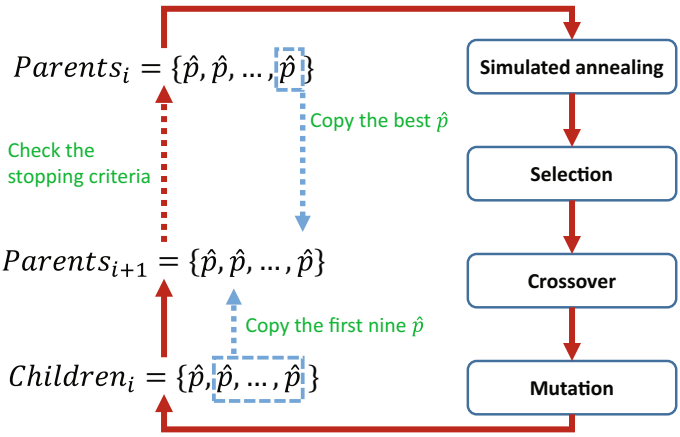


Fig. 2. Process of HSAGA.

In hybrid simulated annealing genetic algorithm, the population is consists of vaccine priority represented in prefix sum format. The population size is ten, and each iteration begins with simulated annealing step to perturb each candidate, followed by selection, crossover and mutation. Figure 2 is the flow chart of the process. For a given allocation, we carried out 5 simulation runs, and the fitness score is the average value of objective function of each run, and the smaller the fitter for both total number of infected cases and economic impact. The best solution of the previous generation and the first nine solutions for this generation become the candidates of next generation. At the beginning of each iteration, we carry out a simulated annealing step for each candidates. It is

a temperature controlled mutation, that is we mutate each candidate according to the temperature (that is number of iterations up to the point in our case). The process stops at 200 iterations and the early stop condition is that five consecutive iterations consist of the same candidates. (the convergence of the stopping criterion discussed in Sect. 3.) The vaccine priority is encoded in prefix sum format, that is $p = (20, 50, 50, 70, 20, 30, 10)$ can be written as $\hat{p} = (20, 70, 120, 190, 210, 240, 250)$, since the total amount of vaccine is always 2.5 millions the last coordinate can be dropped. Given two genes (vaccine priorities), the crossover operation is the following: Randomly generate a pair of numbers g_1, g_2 where $0 \leq g_1 \leq g_2 \leq 250$, if the interval $[g_1, g_2]$ covers the same number of chromosomes, then we exchange the covered part. The segment of chromosomes x_i, x_j is covered by interval $[g_1, g_2]$ if and only if $x_{i-1} \leq g_2 \leq x_i$ and $x_j \leq g_2 \leq x_{j+1}$. Figure 3 is an example of crossover operation. We randomly increase g_1 or decrease g_2 if a direct exchange is invalid, that is the length of covered segments differs.

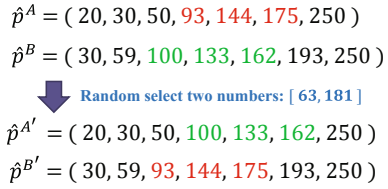


Fig. 3. Crossover of HSAGA.

The mutation operation is defined as following: Randomly pick index i and randomly generate a number x, replace x_i with x and sort the resultant sequence. Figure 4 is an example of mutation operation.

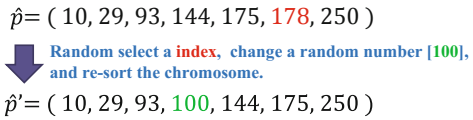


Fig. 4. Mutation of HSAGA.

It is feasible to use simulation results as fitness score, however, the cost can easily become prohibitively high if we allow the search space to include more dimensions, for example, the infectiousness of the virus, p_{trans}. A more efficient approximation function for the fitness score, the surrogate, can speed up the search yet sacrifices accuracy.

We first construct the surrogate for points in which only single age group is vaccinated. That is $p_i = (0, 0, ..., x_i, ..., 0)$. We set our resolution at 100,000, that is the vaccine allocated at 100,000 doses per unit. We carry out simulation at the resolution 100,000, and use linear interpolation to estimate the points not sampled. Note that only a few points are sampled, i.e., simulated, other points are estimated. Let $sim(p_i)$ denote the outcome for all points with only one nonzero dimension. Let $\Delta(p_i)$ denote the value reduced at point p_i, that is $\Delta(p_i) = sim(p_i) - sim(\mathbf{0})$, note that it is always a negative value. Given a point $p = (x_1, ..., x_7)$, the single variable surrogate for p, denoted by $sur_1(p)$, is:

$$sur_1(p) = sim(\mathbf{0}) + \sum_{i=1}^{7} \Delta(p_i) \tag{1}$$

The intuitive explanation is that we can add the contribution of individual age group to be the effect of vaccination priority p.

We plot the results of points simulated for the case that the objective function is total infected cases. In Fig. 5, we can see the results of the case when vaccine efficacy $e = (0.9, 0.9)$. The solid dots are the simulation results, and we use linear interpolation to estimate the values of other points. Every line becomes flat when the amount of vaccine is greater than the population in that age group.

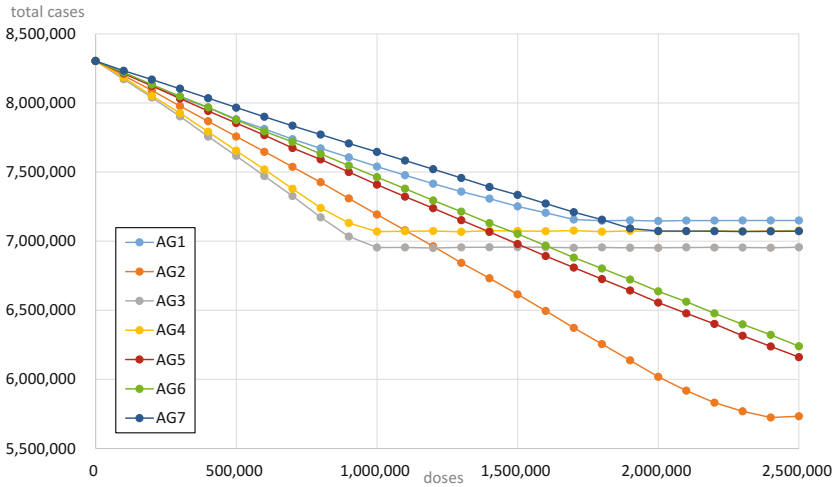


Fig. 5. Simulation points for $e = (0.9, 0.9)$.

When the independent assumption is closer to the reality, the above approximation works better. However, vaccination of one age group do have some effect on other age groups. The interaction among age groups can be intricate. For example, in Fig. 6 the interaction between middle school children and young adults is demonstrated. The the difference between $sim(p)$ and the estimation

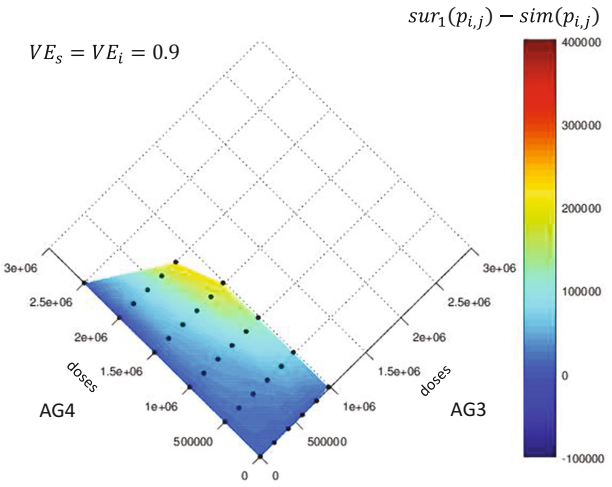


Fig. 6. Error of $sur_1(p)$ for $e = (0.9, 0.9)$.

$sur_1(p)$ increases when the amount of vaccine allocated to these two age groups increases. This demonstrates the idea of herd immunity, that is the individuals get protected when some of their contacts vaccinated.

To study the interaction, we sample some two value points, that is $p_{j,k} = (0, 0, ..., x_j, 0, ..., x_k, ..., 0)$, for each age group we use one fifth of the population as the incremental unit. That is for each age group we try five possible values, called sampled value. There are twenty one combinations of two age group, and for each combination there are twenty five points to be simulated.

We use $\delta(p_{j,k})$ to denote the extra reduction due to interaction. That is the cases saved after individual effects being accounted. If $p_{j,k}$ is a sampled point then $\delta(p_{j,k}) = sim(p_{j,k}) - sur_1(p_{j,k})$, otherwise pick sampled values which are closet lower bound and upper bound of x_j, x_k $a_{j,s}$, say $a_{j,s+1}, a_{k,t}, a_{k,t+1}$, such that $a_{j,s} \leq x_j \leq a_{j,s+1}$ and $a_{k,t} \leq x_k \leq a_{k,t+1}$. The combination of these four values gives us four sampled points, and using a bilinear interpolation we derive $\delta(p_{j,k})$. Given an arbitrary point p, we can define the surrogate to be:

$$sur_2(p) = sim(\mathbf{0}) + \sum_{i=1}^{7} \Delta(p_i) + \sum_{j=1}^{6} \sum_{k=j+1}^{7} \delta(p_{j,k}) \tag{2}$$

3 Results

We develop a method to encode the allocation by gray level to facilitate the further exploration and visualization of the relationship between the structure of the allocations and the final outcomes. One encoding scheme, called *volume scheme*, is to set the color white to denote zero dose and black for 2.5 million

doses. Let x_i be the number of doses for age group i, the gray level is computed by following equation:

$$g_i^{volume} = 255 \times [1 - \frac{x_i}{2.5 \text{ million doses}}] \tag{3}$$

Another encoding scheme, called ratio scheme, is to set the color white to denote zero percent of the age group i vaccinated and black hundred percent and we use $\#AG_i$ to denote the population of age group i. The gray level is computed by following equation:

$$g_i^{ratio} = 255 \times [1 - \frac{x_i}{\#AG_i}] \tag{4}$$

Each age group is then assigned a gray level according to the encoding scheme. We use a line segment with that gray level to represent vaccination level of each age group, as shown in the top half of Fig. 7. The allocation is then represented by stacking the seven line segment vertically (in the middle part of Fig. 7, we put the line segment horizontally). For a set of ordered allocations, the line segment for each allocation is stitched together according to the ordering. The sequence

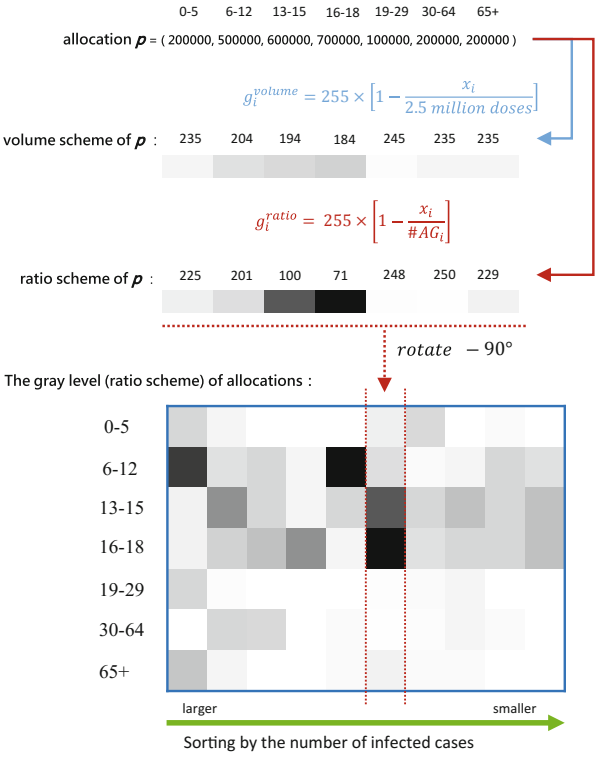


Fig. 7. The gray level [1].

of allocations is sorted from left to right where the better allocations are on the right side.

The rationale of our choice of stopping criteria is explained below. We carried out long testing run with 200 iterations ($HSAGA_{200}$) for $VE_s = VE_i = 0.9$ and minimizing the number of infected cases as the objective. For each iteration we record the best and the worst candidates (*allocations*) in population. As shown in Fig. 8, the best candidate stayed roughly the same after 50 iterations. Therefore, the algorithm stops when all candidates for the last 5 iterations stays the same. The difference between the solutions of $HSAGA$ and $HSAGA_{200}$ is comparable to the coefficient of variation of the simulation system. But the number of allocation examined is reduced from 2,379 to 806.

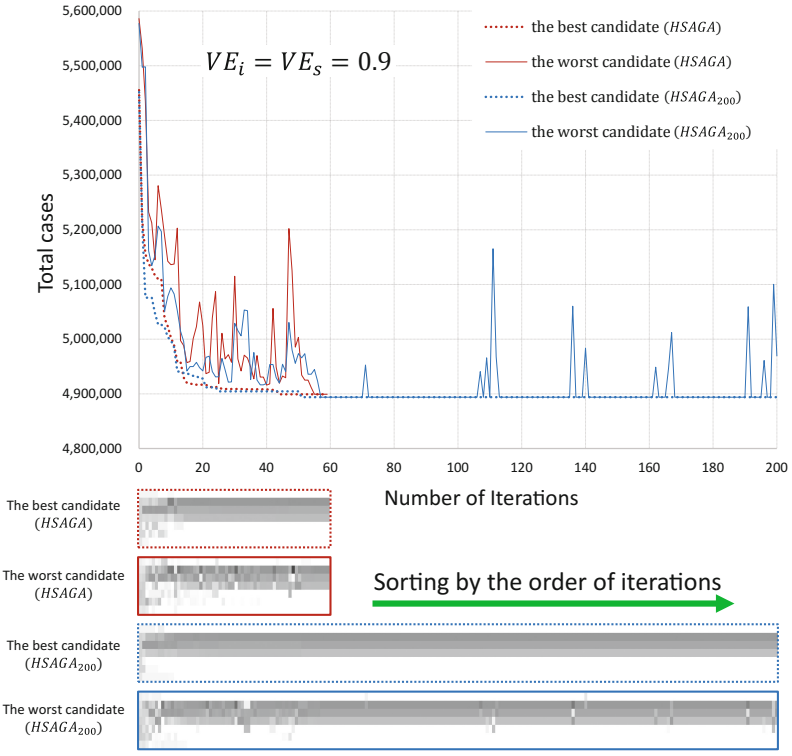


Fig. 8. The best and worst candidates for each iteration [1].

To study the fidelity of surrogates, for each objective, we first define a specific point where every age group is allocated five hundred thousand doses and evaluate this point in twenty five vaccine efficacy scenarios, which are the combination of $VE_i = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and $VE_s = \{0.1, 0.3, 0.5, 0.7, 0.9\}$. The results are summarized in Table 2. It is obvious that when vaccine efficacy increases the

Table 2. Basic data ($p_{trans} = 0.1$).

e	Fitness score	$sim(p)$	$sur_1(p)$	Error(%)	$sur_2(p)$	Error(%)
0.9,0.9	cases	4,455,427	4,788,075	7.466	4,554,759	2.229
	costs	8,310,193,513	9,072,602,444	9.174	8,512,802,617	2.438
0.9,0.7	cases	4,698,266	5,147,952	9.571	4,731,243	0.702
	costs	8,636,331,563	9,528,002,760	10.325	8,710,257,099	0.856
0.9,0.5	cases	4,927,159	5,456,860	10.751	4,993,846	1.354
	costs	8,944,919,355	9,911,405,181	10.805	9,083,785,087	1.552
0.9,0.3	cases	5,136,272	5,704,787	11.069	5,234,124	1.905
	costs	9,224,445,098	10,216,491,911	10.755	9,410,459,583	2.017
0.9,0.1	cases	5,325,681	5,916,227	11.089	5,438,429	2.117
	costs	9,476,035,337	10,475,664,986	10.549	9,682,863,056	2.183
0.7,0.9	cases	4,525,532	4,860,707	7.406	4,600,653	1.660
	costs	8,440,720,497	9,197,907,660	8.971	8,615,053,744	2.065
0.7,0.7	cases	4,923,963	5,344,757	8.546	4,974,683	1.030
	costs	9,044,648,213	9,863,755,496	9.056	9,175,655,156	1.448
0.7,0.5	cases	5,305,347	5,749,844	8.378	5,348,224	0.808
	costs	9,610,802,718	10,414,323,826	8.361	9,709,532,384	1.027
0.7,0.3	cases	5,662,708	6,102,766	7.771	5,719,132	0.996
	costs	10,133,753,596	10,892,462,317	7.487	10,224,617,062	0.897
0.7,0.1	cases	5,989,095	6,396,327	6.800	6,017,286	0.470
	costs	10,605,482,865	11,289,767,047	6.452	10,653,313,538	0.451
0.5,0.9	cases	4,595,987	4,926,931	7.201	4,615,127	0.416
	costs	8,572,456,927	9,316,533,042	8.680	8,628,217,122	0.650
0.5,0.7	cases	5,154,211	5,519,088	7.079	5,217,634	1.231
	costs	9,459,349,619	10,168,897,087	7.501	9,603,219,799	1.521
0.5,0.5	cases	5,696,168	6,054,460	6.290	5,761,392	1.145
	costs	10,295,232,338	10,932,444,351	6.189	10,429,951,232	1.309
0.5,0.3	cases	6,197,872	6,496,988	4.826	6,269,209	1.151
	costs	11,051,416,638	11,560,757,834	4.609	11,188,587,896	1.241
0.5,0.1	cases	6,643,572	6,871,103	3.425	6,686,484	0.646
	costs	11,714,159,054	12,093,787,579	3.241	11,808,407,140	0.805
0.3,0.9	cases	4,667,510	4,986,218	6.828	4,715,911	1.037
	costs	8,706,327,242	9,425,362,495	8.259	8,823,518,764	1.346
0.3,0.7	cases	5,382,617	5,706,862	6.024	5,456,423	1.371
	costs	9,869,728,585	10,497,181,689	6.357	10,031,709,255	1.641
0.3,0.5	cases	6,088,743	6,352,334	4.329	6,146,112	0.942
	costs	10,978,025,225	11,444,599,275	4.250	11,061,241,640	0.758
0.3,0.3	cases	6,729,075	6,904,506	2.607	6,747,887	0.280
	costs	11,958,598,433	12,257,519,318	2.500	11,981,746,073	0.194
0.3,0.1	cases	7,263,008	7,372,166	1.503	7,314,731	0.712
	costs	12,768,149,330	12,944,705,598	1.383	12,876,565,372	0.849

(*continued*)

Table 2. (*continued*)

e	Fitness score	$sim(p)$	$sur_1(p)$	Error(%)	$sur_2(p)$	Error(%)
0.1,0.9	cases	4,736,192	5,055,614	6.744	4,821,752	1.807
	costs	8,834,230,799	9,547,862,072	8.078	9,033,828,853	2.259
0.1,0.7	cases	5,618,656	5,903,184	5.064	5,665,777	0.839
	costs	10,291,768,794	10,839,386,839	5.321	10,384,390,225	0.900
0.1,0.5	cases	6,489,120	6,668,593	2.766	6,492,402	0.051
	costs	11,671,364,522	11,984,186,486	2.680	11,665,366,594	-0.0514
0.1,0.3	cases	7,237,598	7,320,671	1.148	7,259,207	0.299
	costs	12,827,763,376	12,972,558,681	1.129	12,850,427,436	0.177
0.1,0.1	cases	7,818,985	7,838,093	0.244	7,838,812	0.254
	costs	13,721,178,999	13,755,770,229	0.252	13,764,865,999	0.318

Table 3. Runtime with surrogate and simulation.

e	Total infected cases		Total economic impact	
	$T(sim(p))$	$T(sur_2(p))$	$T(sim(p))$	$T(sur_2(p))$
0.9,0.9	82,653.61	53.16	91,518.68	58.28
0.9,0.5	89,637.37	49.65	106,255.28	83.13
0.9,0.1	104,406.32	49.20	106,067.88	55.17
0.5,0.9	91,068.05	57.12	104,915.09	58.73
0.5,0.5	116,980.7	55.30	104,642.72	57.01
0.5,0.1	145,683.09	55.28	92,484.57	50.42
0.1,0.9	99,839.43	47.76	88,817.69	65.37
0.1,0.5	85,584.94	50.45	120,088.66	63.62
0.1,0.1	121,445.46	54.24	76,828.46	80.70

'$T()$': runtime (sec)

number of cases decreases. We define the error to be the difference between the output of the surrogate and the fitness score produced by running simulations divided by the output of simulation. The error of the two variable surrogate is less than 2.3% which is a significant improvement of single variable surrogate which has error rate up to 11% with the objective being minimizing total number of infected cases. For the objective of minimizing total economical impact, the error of two variable surrogate is 2.4 where single variable surrogate has error rate up to 10.8. The improvement testifies that $\delta(p_{i,j})$ captures some interaction between age groups. The supremacy of two variable surrogate is obvious, from now on we only report the results of comparing the two variable surrogate and real simulation.

In Table 3, we summarize the statistics of computational complexity of the two approaches. It clearly demonstrated that time complexity wise, the surrogate

approach is at 1000 times faster than using simulation as fitness function except the case where $e = (0.1, 0.1)$ and the objective is to minimize economical impact.

Next we feed the points selected by $HSAGA$ with surrogate to the simulation program and the results are summarized in Table 4. The errors are all below one percentage and the average of absolute value is 0.253% for total infected cases and 0.216% for economical impact, both are not too far from the stochastic variation, estimated to be 0.2 percent.

Table 4. Best points by surrogate evaluated with simulation.

e	Total infected cases			Total economic impact		
	$sur_2(p)$	$sim(p)$	error(%)	$sur_2(p)$	$sim(p)$	error(%)
0.9,0.9	4,901,232	4,920,204	−0.386	9,460,595,360	9,416,211,685	0.471
0.9,0.5	5,208,127	5,242,480	−0.655	9,938,339,456	9,948,597,768	−0.103
0.9,0.1	5,507,174	5,514,175	−0.127	10,224,255,420	10,218,091,156	0.060
0.5,0.9	5,006,845	5,011,839	−0.100	9,684,640,457	9,646,015,970	0.400
0.5,0.5	5,844,636	5,831,230	0.230	11,019,845,319	11,993,632,032	0.398
0.5,0.1	6,661,038	6,663,824	−0.042	11,989,881,558	11,993,632,032	−0.031
0.1,0.9	5,102,442	5,125,954	−0.459	9,908,066,854	9,893,251,961	0.150
0.1,0.5	6,616,598	6,604,124	0.189	12,239,839,408	12,245,493,581	−0.046
0.1,0.1	7,851,587	7,858,971	−0.094	13,846,741,066	13,886,024,236	−0.283

For genetic algorithms, the rank preserving surrogates are preferred. One metric to measure the fidelity of surrogates is rank correlation coefficient (r_s) [11]:

$$r_s = 1 - \frac{6 \times \sum_{i=1}^{N} (R_A[i] - R_B[i])^2}{N(N^2 - 1)} \tag{5}$$

All the allocations of Tables 7 and 9 evaluated by the simulation program are collected. For each objective function we do the following. For each allocation there are two fitness scores associated with it, one by simulation program and one by surrogate function. Let R_A be the rank by simulation and R_B rank by surrogate. For each objective the rank correlation coefficient of these two sequences for each setting is shown in Table 5. For the objective to minimizing total number of infected cases, the coefficients are all greater than 98% except one at 93%. And for the objective to minimize economical impact, the coefficients are all greater than 94% except one at 69%. It is reasonable to conclude that the surrogate function preserves the ordering well except the odd case of $e = (0.1, 0.1)$ and objective being minimizing economical impact.

The result of $HSAGA$ with simulation as fitness function is shown in Table 6. All the searches end in less than one hundred iteration, and the number of points examined is in the vicinity of one thousand. We note that the best allocations always concentrate on vaccinating students regardless the efficacy of the vaccine

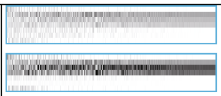
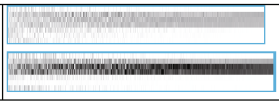
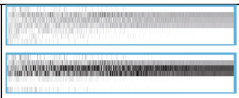
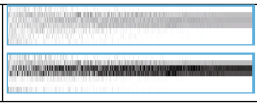
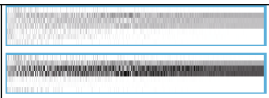
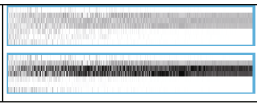
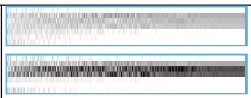
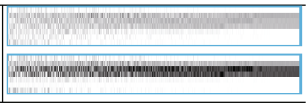
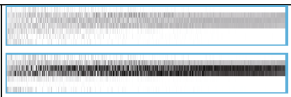
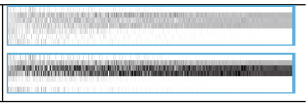
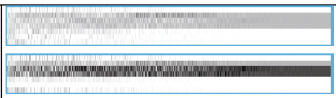
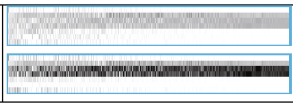
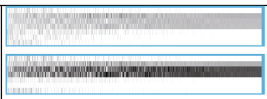
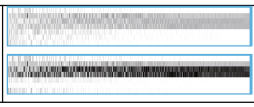
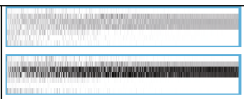
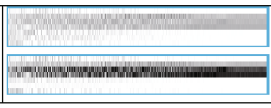
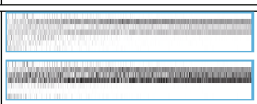
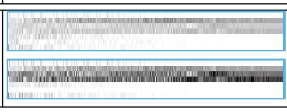
Table 5. Rank correlation coefficient.

e	Total infected cases		Total economic impact	
	N	r_s	N	r_s
0.9,0.9	807	0.99207	1,061	0.98607
0.9,0.5	846	0.99065	1,201	0.95658
0.9,0.1	958	0.99039	1,191	0.95138
0.5,0.9	887	0.98576	1,201	0.98567
0.5,0.5	1,038	0.99072	1,120	0.97843
0.5,0.1	1,207	0.99071	928	0.98363
0.1,0.9	950	0.98516	915	0.97219
0.1,0.5	864	0.99269	1,196	0.94865
0.1,0.1	911	0.93938	702	0.67569

Table 6. The optimal allocation p of $HSAGA$ with total infected cases for each vaccine efficacy.

e	Function	Total cases	Total iterations	Total allocations	p ($\times 10^4$ doses)						
					x_1	x_2	x_3	x_4	x_5	x_6	x_7
0.9,0.9	$sim(p)$	4,899,198	59	806	0	104	79	67	0	0	0
	$sur_2(p)$	4,901,232	72	989	0	89	81	80	0	0	0
0.9,0.5	$sim(p)$	5,212,224	60	846	0	95	83	72	0	0	0
	$sur_2(p)$	5,208,127	68	910	0	80	90	80	0	0	0
0.9,0.1	$sim(p)$	5,490,111	71	958	0	95	78	77	0	0	0
	$sur_2(p)$	5,507,174	69	911	0	80	90	80	0	0	0
0.5,0.9	$sim(p)$	4,978,657	63	887	0	97	80	73	0	0	0
	$sur_2(p)$	5,006,845	82	1,086	0	90	80	80	0	0	0
0.5,0.5	$sim(p)$	5,816,957	79	1,038	0	91	83	76	0	0	0
	$sur_2(p)$	5,844,636	81	1,060	0	88	80	82	0	0	0
0.5,0.1	$sim(p)$	6,644,779	93	1,207	0	83	83	83	1	0	0
	$sur_2(p)$	6,661,038	79	1,047	0	72	88	90	0	0	0
0.1,0.9	$sim(p)$	5,093,340	70	950	0	90	81	79	0	0	0
	$sur_2(p)$	5,102,442	67	901	0	79	88	83	0	0	0
0.1,0.5	$sim(p)$	6,588,382	64	864	0	70	93	87	0	0	0
	$sur_2(p)$	6,616,598	71	964	0	79	81	90	0	0	0
0.1,0.1	$sim(p)$	7,853,438	68	911	0	120	51	79	0	0	0
	$sur_2(p)$	7,851,587	80	1,025	0	100	70	80	0	0	0

Table 7. The gray level of total allocations of $HSAGA$ (total infected cases).

e	total $sim(p)$ in $HSAGA$	total $sur_2(p)$ in $HSAGA$
0.9,0.9		
0.9,0.5		
0.9,0.1		
0.5,0.9		
0.5,0.5		
0.5,0.1		
0.1,0.9		
0.1,0.5		
0.1,0.1		

for the objective to minimize total number of infected cases. The similar conclusion can be made for the case to minimize economical impact, except that when the vaccine efficacy increases some vaccine allocated for elementary students are relocated to young adults.

For a given vaccine efficacy setting, the $HSAGA$ examined around one thousand vaccine allocations. These allocations are sorted according to their fitness score and the sequence is visualized according to the method above shown in Table 7. The sorted sequence for each setting is visualized with volume scheme, the top one, and with ratio scheme, the bottom one. We can see that for those allocations on the right end, the black segments are concentrating on school children. And according to those bottom graphs junior high and high school

students get the highest priority. More specifically, for 2.5 million doses, 70 to 90 percent of junior high and high school students get vaccinated and the rest goes to elementary school students.

The same $HSAGA$ process is carried out with surrogate in place of the simulation and the results are summarized in Table 6. The visualization is shown in Table 7. It is clear that the general recommendation is also to vaccinate school children.

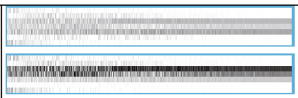
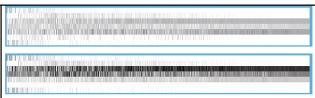
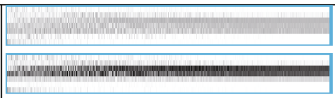
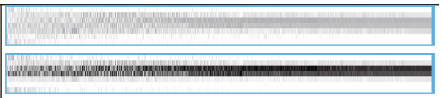
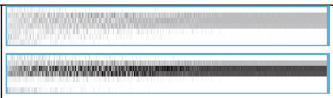
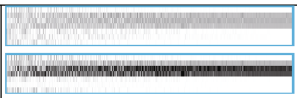
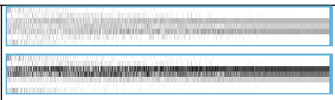
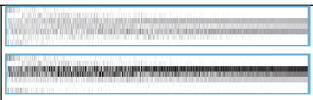
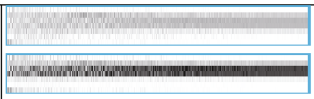
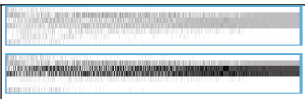
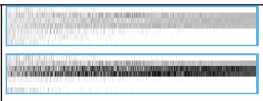
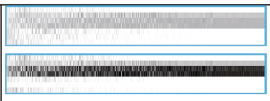
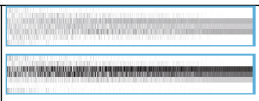
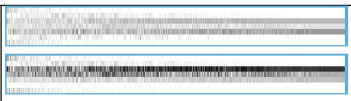
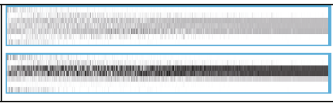
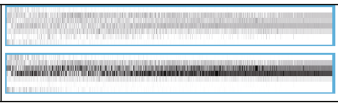
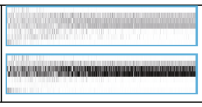
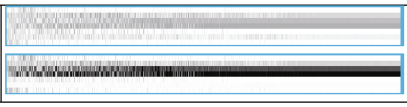
Table 8. The optimal allocation p of $HSAGA$ with total economic impact for each vaccine efficacy.

e	Function	Total costs (US$)	Total iterations	Total allocations	p ($\times 10^4$ doses)						
					x_1	x_2	x_3	x_4	x_5	x_6	x_7
0.9,0.9	$sim(p)$	9,399,213,853	80	1,061	0	0	90	69	91	0	0
	$sur_2(p)$	9,460,595,360	82	1,128	0	0	89	60	101	0	0
0.9,0.5	$sim(p)$	9,903,198,767	83	1,201	0	12	85	68	85	0	0
	$sur_2(p)$	9,938,339,456	118	1,582	0	30	90	80	50	0	0
0.9,0.1	$sim(p)$	10,174,555,653	86	1,191	1	84	82	78	2	3	0
	$sur_2(p)$	10,224,255,420	77	1,062	0	80	90	80	0	0	0
0.5,0.9	$sim(p)$	9,612,330,559	89	1,201	0	0	85	65	100	0	0
	$sur_2(p)$	9,684,640,457	82	1,121	0	0	90	60	100	0	0
0.5,0.5	$sim(p)$	10,956,501,348	77	1,120	0	56	85	80	29	0	0
	$sur_2(p)$	11,019,845,319	82	1,091	0	87	81	82	0	0	0
0.5,0.1	$sim(p)$	11,966,458,234	68	928	0	95	77	78	0	0	0
	$sur_2(p)$	11,989,881,558	71	960	0	70	90	90	0	0	0
0.1,0.9	$sim(p)$	9,836,908,825	64	915	0	0	86	62	102	0	0
	$sur_2(p)$	9,908,066,854	94	1,260	0	0	83	39	128	0	0
0.1,0.5	$sim(p)$	12,188,514,945	89	1,196	0	1	81	82	86	0	0
	$sur_2(p)$	12,239,839,408	92	1,213	0	68	60	80	42	0	0
0.1,0.1	$sim(p)$	13,853,884,964	51	702	4	75	69	92	1	4	5
	$sur_2(p)$	13,846,741,066	107	1,467	0	48	80	96	0	20	6

We first search good vaccination strategy using $HSAGA$ with computer simulation as the fitness scorer. The results are shown in Table 8. We again use the grey level plot to visualize the good strategy for different vaccine efficacy. The result is shown in Table 9, where the economical impact decreases from left to right. In Table 8, it is still the case that the best strategy is to vaccine school children especially junior high and high schoolers. However, when VE_s increases, the vaccine allocates more to young adults and less to elementary school children.

The efficacy of the vaccine is difficult to determined beforehand. Although, we searched for best allocation for each vaccine efficacy setting. It is desirable to know if vaccine efficacy has a big impact on the choice of vaccine allocation. It is clear that the qualitative statement, "vaccinate school children", applied to all scenarios. We compute one specific allocation, $p^{sp}_{case} = (0, 900000, 800000, 800000, 0, 0, 0)$, for all scenarios, and compare with the solutions produced by $HSAGA$ with surrogate (Table 6). The result is show in Table 10. Since the standard deviation of the simulation system is $10,850$ and the coefficient of variation is about $0.2\,\%$, it is not too stretchy to say that

Table 9. The gray level of total allocations of $HSAGA$ (total economic impact).

e	total $sim(p)$ in $HSAGA$	total $sur_2(p)$ in $HSAGA$
0.9,0.9		
0.9,0.5		
0.9,0.1		
0.5,0.9		
0.5,0.5		
0.5,0.1		
0.1,0.9		
0.1,0.5		
0.1,0.1		

allocation p^{sp}_{case} works well even if we do not know the efficacy of the vaccine. For the objective to minimizing total economical impact, we simply allocate more vaccine to junior high school and high school students and allocate the rest of stockpile equally to elementary school students and young adults, $p^{sp}_{cost} = (0, 450000, 800000, 800000, 450000, 0, 0)$ is the result. For p^{sp}_{cost}, the error is less than 1.63%, although it is much higher than 0.2%, but still a good enough under the uncertainty of vaccine efficacy (compare with Table 8).

Table 10. Impact of vaccine efficacy.

e	Total infected cases			Total economic impact		
	$sur_2(p^{sp}_{case})$	Difference	error(%)	$sur_2(p^{sp}_{cost})$	Difference	error(%)
0.9,0.9	4,900,564	−668	−0.014	9,597,357,275	136,761,915	1.446
0.9,0.5	5,222,468	14,341	0.275	9,962,001,314	23,661,858	0.238
0.9,0.1	5,509,732	2,558	0.046	10,346,550,571	122,295,151	1.196
0.5,0.9	5,006,845	0	0.000	9,842,475,757	157,835,300	1.630
0.5,0.5	5,844,413	−223	−0.004	11,071,276,794	51,431,475	0.467
0.5,0.1	6,674,396	13,358	0.201	12,116,240,881	126,359,323	1.054
0.1,0.9	5,097,761	−4,681	−0.092	10,003,117,556	95,050,702	0.959
0.1,0.5	6,618,829	2,231	0.034	12,268,792,069	28,952,661	0.237
0.1,0.1	7,862,525	10,938	0.139	13,894,648,482	47,907,416	0.346

4 Conclusion and Discussion

Our results confirm the finding of previous studies that school children should be vaccinated with high priority no matter in term of the number of total infected cases or economical impact. We discover that a good allocation for one specific vaccine efficacy setting works well for other settings as well. Although the preliminary results are promising, a thorough study with parameters such as transmission probability as well as household structures is necessary before a definite conclusion can be drawn. We also notice that certain pair of age groups has stronger interaction, that is their collective protection is much stronger than the sum of individual protections. We suspect that the connection patterns of the underlying contact network implicitly defined in the simulation play an important role.

We propose to use surrogate-based evolution computation to search the vast scenarios of agent-based stochastic disease spreading simulation. Our surrogate construction scheme works for both effective measurements. The average of error of two variable surrogate is less than 0.3%, when the objective is to reduce the total number of infected case; it is less than 0.5% with total economical impact as the objective.

We note that certain age group combination has stronger interaction, that is their collective protection is much stronger than the sum of individual protections. And we suspect the connection patterns of the underlying contact network implicitly defined in the simulation play an important role.

One obvious future direction is to explore the vast landscape of scenarios with various objective functions and constraints. For example, the vaccine available date may vary, the infectiousness of the virus strand might vary, and other mitigation strategies such as antiviral treatment and school closure might vary. The objective function can vary too.

Currently, we construct our surrogate using only the output of the simulation results. However, the intrinsic structure used by the simulation program might be useful information to construct more efficient and higher fidelity surrogate. Moreover, mathematical diseases modes might provide important insight for this direction.

Finally, we envision that an autonomous software searches through the huge scenario space with the help of surrogate function and adaptively executes simulation program to revise the surrogate function to produce higher fidelity surrogate and better search results.

References

1. Jian, Z.-D., Hsu, T.-S., Wang, D.-W.: Searching vaccination strategy with surrogate-assisted evolutionary computing. In: Proceedings of SIMULTECH 2016 (2016)
2. Chang, H.-J., Chuang, J.-H., Fu, Y.-C., Hsu, T.-S., Hsueh, C.-W., Tsai, S.-C., Wang, D.-W.: The impact of household structures on pandemic influenza vaccination priority. In: 5th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, pp. 482–487 (2015)
3. Basta, N.E., Halloran, M.E., Matrajt, L., Longini, I.M.: Estimating influenza vaccine efficacy from challenge and community-based study data. Am. J. Epidemiol. **168**(12), 1343–1352 (2008)
4. Meltzer, M.I., Cox, N.J., Fukuda, K.: The economic impact of pandemic influenza in the United States: priorities for intervention. Emerg. Infect. Dis. **5**(5), 659–671 (1999)
5. Jin, Y.: Surrogate-assisted evolutionary computation: recent advances and future challenges. Swarm Evol. Comput. **2**(1), 61–70 (2011)
6. Grefenstette, J.J., Fitzpatrick, J.M.: Genetic search with approximate fitness evaluations. In: International Conference on Genetic Algorithms and Their Applications, pp. 112–120 (1985)
7. Lee, Y.B., Brown, T.S., Korch, W.G., Cooley, C.P., Zimmerman, K.R., Wheaton, D.W., Zimmer, M.S., Grefenstette, J.J., Bailey, R.R., Assi, T.-M., Burke, S.D.: A computer simulation of vaccine prioritization, allocation, and rationing during the 2009 H1N1 influenza pandemic. Vaccine **28**(31), 4875–4879 (2010)
8. Tsai, M.-T., Chern, T.-C., Chuang, J.-H., Hsueh, C.-W., Kuo, H.-S., Liau, C.-J., Riley, S., Shen, B.-J., Shen, C.-H., Wang, D.-W., Hsu, T.-S.: Efficient simulation of the spatial transmission dynamics of influenza. PLoS ONE **5**(11), 1–8 (2010)
9. Germann, T.C., Kadau, K., Longini, I.M., Macken, C.A.: Mitigation strategies for pandemic influenza in the United States. Proc. Natl. Acad. Sci. **103**(15), 5935–5940 (2006)
10. Fu, Y.-C., Wang, D.-W., Chuang, J.-H.: Representative contact diaries for modeling the spread of infectious diseases in Taiwan. PLoS ONE **7**(10), 1–7 (2012)
11. Loshchilov, I., Schoenauer, M., Sebag, M.: Comparison-based optimizers need comparison-based surrogates. In: Parallel Problem Solving from Nature, PPSN XI: 11th International Conference, Kraków, Poland, 11–15 September 2010. Part I, pp. 364–373 (2010)

Modelling Population Dynamics Using a Hybrid Simulation Approach: Application to Healthcare

Bożena Mielczarek(✉) and Jacek Zabawa

Department of Operations Research, Faculty of Computer Science and Management, Wrocław University of Science and Technology, Wrocław, Poland
{Bozena.Mielczarek,Jacek.Zabawa}@pwr.edu.pl

Abstract. The goal of the study is presenting a population submodel developed using the system dynamics (SD) approach and discussing solutions for the integration of the SD methodology with discrete time control in formulating long-term projections for population evolution and its influence on healthcare demand. This study relies on historical demographic data and officially formulated scenarios for the most likely population projections for the Wrocław Region. The historical parameters are applied from 2002 to 2014, and projected trends are adopted for 2015 to 2035. The preliminary findings confirm the validity of using the hybrid simulation approach for a more advanced exploration of demography-dependent health policy issues.

Keywords: System dynamics · Discrete simulation · Demography · Age pyramid · Healthcare demand

1 Introduction

Economic studies typically depend on extended demographic forecasts, and population projections are an essential and imperative input for a range of such analyses. Credible estimations on the size and structure of future population directly affect the correct examination of long-term macroeconomic performances and behaviours. Therefore, economic analyses that consider the effects of population dynamics usually take into account demographic forecasts. For example, the impact of population aging on economic growth and the macro-economy in general is one of the most common issues to be researched in economics [1]. As such, the number of elderly citizens and the share of insured employees among the entire adult population influences the financial sustainability of the social security system [2]. Moreover, on-going changes in the structure of local populations affect urban development and land use [3]. Health policy models use population projections as important key inputs, equally essential to epidemic data [4].

The most common approach to incorporating population forecasts in economic studies is based on using census data and moderate demographic scenarios developed and published by national statistical units or global organizations such as the World Bank or the United Nations. Another popular solution for predicting population

M.S. Obaidat et al. (eds.), *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing 676,
https://doi.org/10.1007/978-3-319-69832-8_14

changes is to utilize a stochastic forecasting method. For example, time series modelling techniques are frequently applied when estimating basic demographic parameters such as fertility, mortality, and migration rates [5]. In particular, short-term fluctuations in vital statistics are adequately modelled with this well-developed methodology. As such, Lassila et al. [6] proposed the original method of stochastic forecasting with revisions of demographic forecasts. The primary concept of their approach is that each update in the official population projection alters people's perception of the future.

Another approach that successfully addresses the age-structured demographics in many real-life systems is simulation modelling. Assuming that basic demographic measures, such as fertility, mortality, and migration, may be considered as descriptive parameters of stochastic processes, the stochastic simulation may be applied to analyse aspects on demographic uncertainty. First, the predictive distributions of future population are formulated and, utilizing Monte Carlo (MC) methods, the output estimations of population structures are obtained [2]. The MC approach may also be supported with micro-simulations [7] that enable to model individual behaviour. The MC processes are then used to convert global probabilities into characteristics of individual behaviour.

Another well-known simulation methodology offers an alternative perspective on demographic phenomena. The system dynamics (SD) approach proved to be useful in policy formulation and for addressing the dynamic complexity of a system [8]. For example, Barber and Lopez-Valcarcel [9] used an SD submodel to simulate demographic changes towards forecasting the demand for medical specialties. Masnick and McDonnel [10] used the SD approach to link groups of individuals with health conditions to clinical workloads.

The aforementioned simulation approaches differ significantly. An MC model simulates a range of potential scenarios, each with an assigned probability of occurrence, and produces forecasts, usually in the form of relevant means, probabilities, and a dispersion of results around an expected value. The SD is a deterministic approach, particularly helpful when the goal of a study is to formalize a mental model of a complex phenomenon. It also helps explore the relationship between a system's structure and its behaviour after minor or major changes. Typically, SD models are not created to yield exact quantitative predictions, but are intended to explore multiple policy options. These models may enable a better understanding of the performance of large and complex systems, allowing for both qualitative and quantitative problem analyses.

The overall goal of our project is to develop a methodology that would enable the prediction of the future demand volume and intensity for healthcare services. Consequently, we aim at the development of a hybrid simulation model that would allow alignment of demographic forecasts with health policy models. To this effect, we combine the discrete-event simulation (DES) approach, the most often used technique in the field of healthcare management [11, 12], with an SD paradigm and build a hybrid simulation model. This would enable us to preserve the unique and valuable features of the DES approach with the possibility of holistically analysing the problem according to SD methodology. In this study, we focus on presenting the SD population submodel driven by the separate discrete engine that controls the frequency of near continuous computation and shifts the members of a population between successive age cohorts. The hybrid model performs time-step simulations that replicate population evolution

according to the continuous SD paradigm, while the simulated demographic changes directly influence the DES submodel that generates the discrete demand for healthcare services.

The input parameters describing the population are calculated based on historical and forecasted rates of primary demographic parameters, retrieved from various databases and official projections published by the Polish Central Statistical Office (CSO) [13]. The output of the simulation is represented by the total number of individuals in every age/gender cohort. The model enables us to distinguish each individual in every cohort group. The movement of individuals with health conditions are recorded after the patient is first registered in the healthcare system. The model samples the attributes from empirical distributions, and adjusts and assigns them individually to each patient.

In our research, we expect to verify the credibility of the approach based on the SD method, developed by Forrester [14] and extended by a modification of time step in the population module in response to feedback from the discrete module.

2 Healthcare Simulation

According to many authors, simulation plays an important role in healthcare decision making [15]. This commonly used decision support technique allows health policy makers to examine short- and long-term effects of planning processes and determine the implications of prevention and treatment procedures at regional and national levels. Overall, simulation modelling helps analyse the current work of healthcare service providers. Studies typically concentrate on the unit providing healthcare services, such as hospitals, operating theatres, outpatient departments, emergency units, or diagnostic centres [16, 17]. Models for simulating epidemics are designed to predict the dynamic rate and spread of infectious diseases, and analyse the direct and indirect causes of the intensification of civilizational diseases (e.g. dementia, diabetes, cardiovascular diseases) [18, 19]. As such, simulations are used to forecast long-term population needs and determine the resources needed to cover the expected demand [20, 21]. Simulation methods are helpful when evaluating the plans for rescue activities in extreme situations, such as natural disasters, traffic problems, industrial incidents, and terrorist or bioterrorist attacks [22, 23]. Simulation is also a popular educational tool and a widely applied approach in a range of research studies.

Many applications of simulation in healthcare have been demonstrated in past decades. However, a universal, conclusive procedure for matching the most suitable simulation technique to a specific problem has not been hitherto elaborated. According to many authors, the DES approach is the most popular technique in the field of healthcare management [11, 12]. However, within the area of health policy and forecasting, when research aims at establishing long-term predictions for health service demand, the SD approach is also a frequently selected method. Health policy studies are usually strongly associated with the structure of the population and its dynamics on a local, regional, or national level. The intensity and diversity of population needs strongly depend on age-gender cohorts that, in turn, change according to constantly observed variations in such demographic parameters, such as the average expected

length of life, birth and death rates, and fluctuations in migration parameters [4]. The amount of time that passes from the moment a diagnosis is formulated further influences the level and structure of healthcare needs for patients with diagnosed diseases [24].

According to [25], when using the SD approach, we gain a systematic view of patient movements and a more strategic perspective of system behaviour, although we lose the ability to include uncertain factors that are prominent in healthcare systems. Additionally, patient-oriented issues are considered as aggregate, instead of individual level.

In this study, we used the well-established methodology of SD modelling to capture overall population evolution. However, in order to be able to maintain the uncertain character of system behaviour, we modified the approach to allow for the use of age-gender cohorts simulated by the SD submodel as to generate discrete demand for healthcare services.

3 Input Data

The study was conducted in the two subregions of Lower Silesia, the fourth largest region in Poland. These two subregions, named *the Wrocław Region* (WR), after the capital of Lower Silesia (Wrocław), and eight other administrative districts situated nearby the capital, are the subject of our analysis.

According to the CSO [13], total WR population has been increasing annually since 1995 (Table 1, Fig. 1) for both genders. Between 1995 and 2014, the male population increased by 3.18% and the female population by 4.87%. This growth is a result of the aging trend.

Table 1. Structure of the WR population according to age-gender groups from 1995 to 2014.

	1995	2000	2005	2010	2014
Total number of females (F) and males (M)					
F	604848	604226	608160	622112	633074
M	561852	557928	558057	570442	579707
Children aged 0–4 as % of total number of females/males					
F	5.05%	4.19%	3.87%	5.03%	4.82%
M	5.72%	4.79%	4.45%	5.74%	5.62%
Children aged 5–19 as % of total number of females/males					
F	21.72%	19.60%	15.70%	13.21%	12.88%
M	23.38%	21.23%	17.11%	14.40%	14.06%
People aged 60+ as % of total number of females/males					
F	19.16%	20.40%	20.79%	23.38%	26.17%
M	13.57%	14.05%	14.19%	16.84%	19.52%

Source: [13]

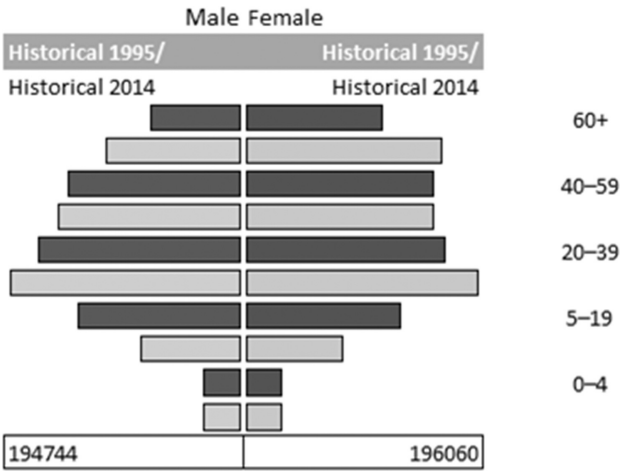


Fig. 1. Comparison of age pyramids of the WR population using historical data from 1995 (dark colour) and 2014 (light colour). The lengths of the horizontal bars correspond to cohort size. The largest cohorts are displayed at the bottom of the graph (Source: [26]).

From among the two edge groups, the youngest population in relation to the total population decreases, while the oldest population expands. For example, the number of adolescent cohorts (females and males between 5 and 19 years old) decreased by 37.93% (girls) and 38.37% (boys) between 1995 and 2014, while the number of senior cohorts increased by 42.94% (females) and 48.40% (males) during the same period.

4 Model Description

We have applied the chronological ageing approach described by Eberlein et al. [27]. The outline of the first version of the model was presented in [28] and the extended version was described in [26]. To better visualize the general concept, we present population aging chains using SD notations (Fig. 2).

There are ten cohorts and ten state variables that define cohort population: five female and five male cohorts. Each cohort represents a separate state variable described by the stock level. The stocks accumulate dynamic objects that move through the system. At each simulation time step, the stocks report the present quantitative status of the individuals belonging to particular cohort. At each moment of passing simulation time, the stocks report the present quantitative status of the individuals belonging to particular cohort. In accordance with Krahl [29], we defined the internally generated state-change events and linked them with state variables. The state variables change at discrete times when their associated flow rates also change. The flows are used to model the movement of individuals over a specified period of time. Input and output flows that move to and from the particular stock are aggregated into one dynamic object that controls the appropriate state variable. The resultant flow instantly increases or decreases the number of individuals in the cohort. This eliminates rounding errors and

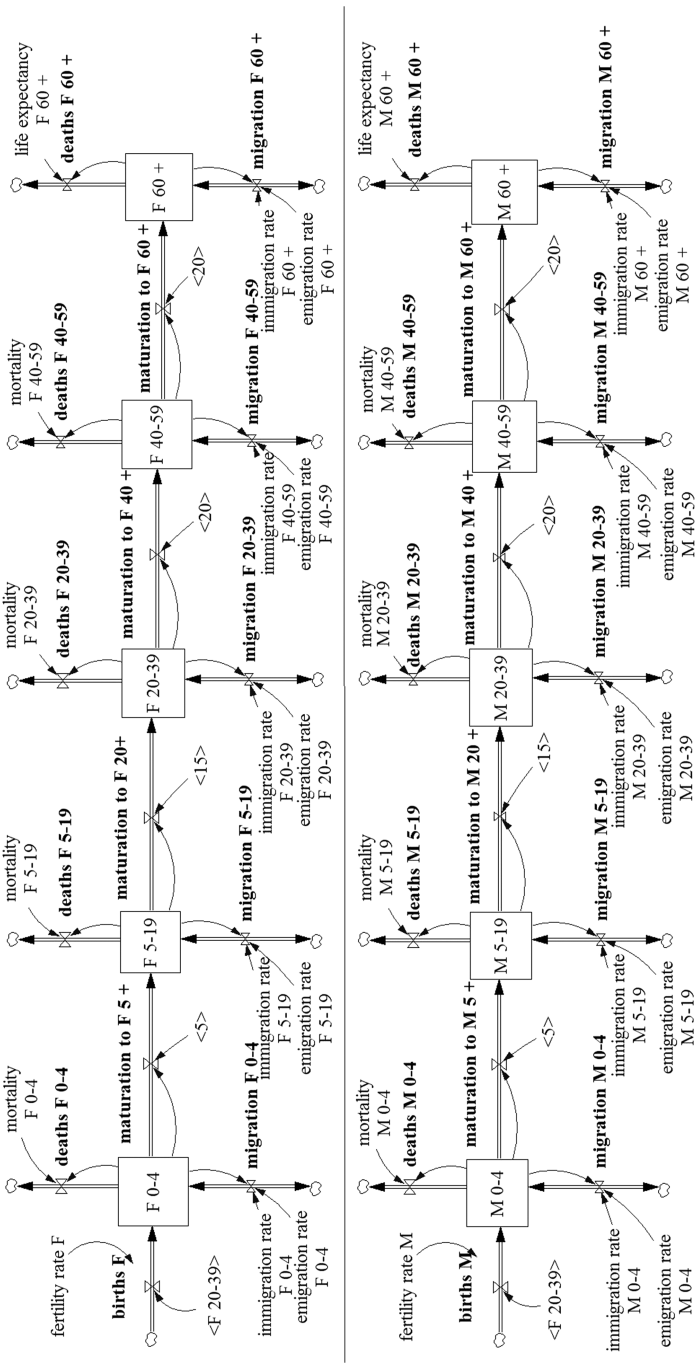


Fig. 2. Population aging chains (Source: [26]).

significantly improves accuracy of the simulation output. The initial population data matches historical conditions in 2002 based on information published by the CSO [13].

The simulation begins in 2002 and runs through 2014 according to parameters calculated on the basis of historical values extracted from statistical data bases for the WR. Beyond 2014, the exogenous parameters are extrapolated based on the official forecasts published by the CSO [30]. All input parameters are calculated separately for female and male cohorts and for each calendar year, according to the simulation horizon. The solid empirical grounds of our model increase the credibility of the simulation.

4.1 Computer Model

The ExtendSim [29] environment was chosen to construct the model due to the unique capabilities of the software to link two simulation approaches in one construct. There are two submodels that work simultaneously and exchange information on a regular basis: the DES submodel, which was built using modules from the Item library, and the SD submodel, developed from the blocks of the Value library. Both libraries are available in the standard software package. The model uses a number of integrated blocks defined according to the hierarchic approach.

The SD submodel represents the demographic dimension of the model, and simulates the dynamic changes in the WR population. The entire population is divided into ten main integrated blocks representing ten age-gender cohorts (see Fig. 2). A crucial role in the hierarchic SD blocks is played by the *Holding Tanks*—the elementary blocks representing the stocks of the SD approach. The DES submodel represents the healthcare aspects of the model, and generates patient arrivals to the healthcare system. The DES submodel consists of the integrated block that simulates the prevalence of needs-for service events. The two submodels are integrated and communicate with each other: the SD simulates on-going changes in the WR population and the DES generates the demand based on information passed from the age-gender cohorts. The additional DES module helps control the passage of time in both submodels. Consequently, the control of the SD objects is overtaken by the discrete blocks.

The built-in mechanism for database management is applied to enable storing all input parameters and output simulation data in the external databases.

4.2 Cohorts 0–4

The youngest population is described by two cohorts, F 0–4 and M 0–4 (Fig. 3), separately for females and males. Both youngest cohorts are affected by two input and three output flows. There is one primary and one additional input flow: births and immigration, and one primary and two additional output flows: maturation, deaths, and emigration.

The number of females of childbearing age (F 20–39 cohort) influences the primary input flow (births) for both females and males. The initial values for fertility rates (from 2002 to 2014) are calculated based on data published by CSO [13]. Particularly, female

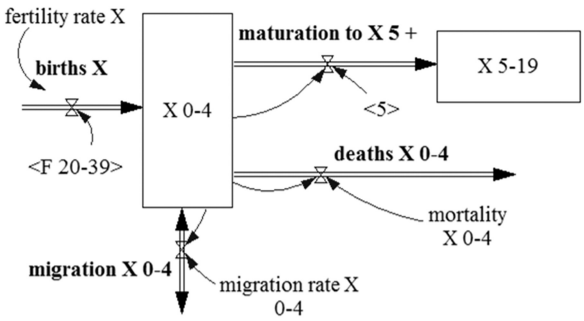


Fig. 3. The youngest cohorts. X denotes F (female) or M (male).

fertility rates for each historical year are the result of dividing the total number of girls aged 0–4 by the total number of females aged 20–39. Accordingly, male fertility rates are calculated by dividing the total number of boys aged 0–4 by the total number of females aged 20–39.

The primary output flow (maturation) describes the basic outflow of individuals from the cohort. It is interpreted as the average residence time needed for a child to leave the youngest cohort and enter the subsequent one (i.e. cohort 5–19). The values of maturation time differ between every pair of cohorts. For example, it is five years between F 0–4 and F 5–19, but 20 years between M 40–59 and M 60+. The immigration and emigration input and output flows (migration) depend on migration rates for young children (ages 0–4) moving to and from the WR and the total number in the youngest cohort. The rates are calculated separately for boys and girls. The deaths output flow is driven by death rates, calculated based on the number of recorded deaths among the youngest WR population and the total number of children aged 0–4 living in WR, separately for females and males.

The input parameters for 2002–2014 are calculated on the basis of historical values, extracted from statistical databases for the WR. Beginning in 2015, the hypothetical values of female and male fertility rates, migration rates, and death rates are adopted according to different scenarios for population projections [30]. This is further described in Subsect. 7.1.

4.3 Cohorts 5–19, 20–39, and 40–59

The series of middle cohorts, 5–19, 20–39, and 40–59, describes the WR population older than four and younger than 60, separately for females and males. The cohorts are affected by three input and two output flows, defined similarly to those of the youngest cohort 0–4 (Fig. 4). The main difference is in the primary input flow, where maturation from the previous cohort is used instead of birth input flow.

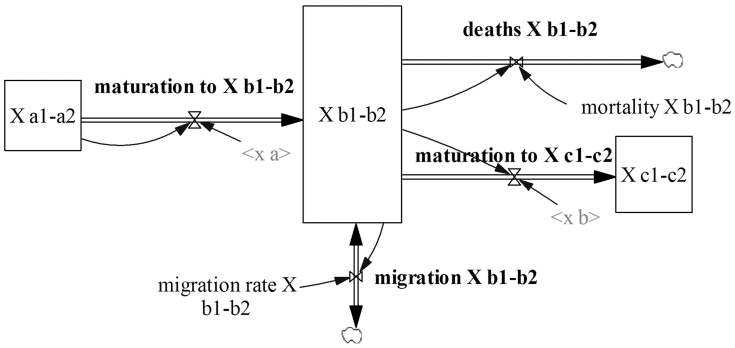


Fig. 4. The youngest cohort. X denotes F (female) or M (male); the meaning of the symbols is as follows: a1–a2 is the previous cohort (e.g. 0–4), b1–b2 is the current cohort (e.g. 5–19), c1–c2 is the subsequent cohort (e.g. 20–39), xa is the maturation time from the previous cohort to the current cohort (e.g. 5 years from the cohort 0–4 to 5–9), and xb is the maturation time from the current cohort to the subsequent cohort (e.g. 15 years from the cohort 5–19 to 20–39).

4.4 Cohort 60+

The last two cohorts, F 60+ and M 60+, describe the oldest population, separately for females and males (Fig. 5). The oldest cohorts are influenced by two input and two and output flows: maturation from the previous cohort, migration (immigration and emigration), and deaths. The flows are defined similarly to those of the youngest cohort 0–4, except for the last one. The primary output flows (deaths) use the average life expectancy parameters for female(s)/male(s) at the age of 60, instead of the death rates. Historical values of these parameters (from 2002 to 2014) are estimated based on data published by the CSO [13]. From 2015, the hypothetical values of female and male average life expectancy are adopted according to different scenarios of population projections, as published in [30]. This is further described in Subsect. 7.1.

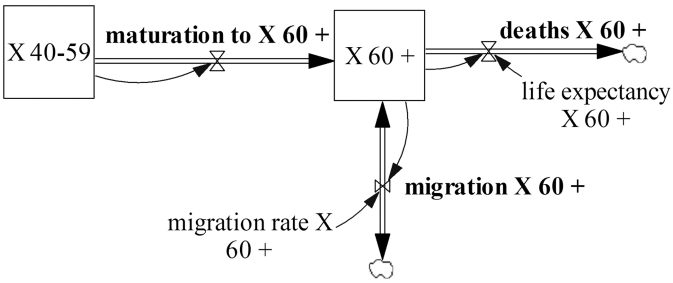


Fig. 5. The oldest cohort. X denotes F (female) or M (male).

4.5 SD-DES Time Mechanism

The unique feature of the SD modelling approach is that numerical calculations are based on differential equations. This enables the continuous observation of the behaviour of the dynamic process and allows the simulation to be designed for a continuously changing system. In demographic studies, however, it is more convenient to capture key events, such as births and deaths, at discrete moments, thus making the mixed technique of transferring elements from one cohort to another at certain moments and registering on-going changes in population structure on a continuous basis more appropriate.

We have implemented the time-step mechanism, which is designed with the assumption that time passes according to small constant discrete values. The flow rates change at discrete moments defined by the time step, and values of all input and output flows are updated according to differential equations. The flows that are connected with particular stocks are subsequently aggregated into one dynamic object. The resultant flow instantly increases or decreases the number of individuals in the cohort, and the new value of stock level is registered. This approach may be described as the sampling procedure that reads the value of the stock level, updates this value based on information from the resultant flow, and downloads the obtained values into separate objects, that is, the holding tanks. The key feature of this approach is the ability to memorize, for any simulation step, not only the number of people belonging to a particular cohort, but also their individual attributes, such as age, sex or other attributes assigned to the moving objects (patients).

This ability to preserve the attributes is an extremely valuable quality when attempting to link the SD and DES approaches. As such, the values uploaded from the holding tanks may be used to parameterize the inter-arrival time distributions that describe the patient's presentations to the healthcare system and create the demand for healthcare services. The discrete objects (patients) generated from the random varieties retain their previously acquired attributes and may enter the DES model without delay.

5 Blending Problem

One of the challenges to be solved when using the SD approach to model the chronological aging of the population is the blending problem [27]. Using a series of stocks to represent population, the level of the population in every cohort and the total one will be smaller than expected if there are intermediate outflows that drain the cohorts located in the middle of the chain. Eberlein et al. [27] proved that if this drainage rate is uniform along the aging chain. However, the total population will not change, and the number of individuals in the older cohorts will be smaller for smaller sizes of the age groups. When the drainage rates are different for different age groups, both total population and that in particular cohorts changes as cohort size changes.

The blending problem is particularly evident when the aging chain is defined for a long-run horizon and for demographic studies. In our model, the cohort blending problem is a troubling issue because every cohort in our aging chain is characterized by

different internal time ranges (i.e. 5, 15, 20 years) and is drained by two intermediate outflows (i.e. deaths and emigration).

To overcome the blending problem, we have applied an optimization algorithm. We adjust maturation times so that the total population level and the number of individuals in every cohort would be similar to the historical data extracted from empirical files for 2002–2014.

5.1 Optimization Algorithm

Optimization, sometimes known as goal seeking, is a useful technique that helps determine ideal values for input parameters. The model is run multiple times using different values for selected input and the solution space is searched until an acceptable solution set is found. The objective function, defined by the user, minimizes or maximizes the pre-defined output measure and, through the process of averaging the samples and sorting the solutions, the best solution set of parameters is found. This set is then used to search for slightly different but possibly better solutions. Every new set of the potential best solution is called a generation. This evolutionary algorithm continues to look for new generations until the probability of finding a better solution is minimal. Subsequently, the process is terminated and the model is populated with the best solutions found during optimization.

The entire procedure comprises ten steps, as presented in Table 2. The sequence of the successive steps is arbitrary and another scheme would probably give other values for the final maturation times.

Table 2. Steps of the optimization process. The search for the best maturation times.

Step	Pair of cohorts	Tested range (years)	Theoretical value (years)	Best solution (years)
The search for the optimized maturation times between every pair of cohorts				
No 1.	F 0–4 and F 5–19	4–10	5	8.1
No 2.	F 5–19 and F 20–39	10–20	15	13
No 3.	F 20–39 and F 40–59	19–40	20	34
No 4.	F 40–59 and F 60+	19–40	20	27
No 5.	M 0–4 and M 5–19	4–10	5	7
No 6.	M 5–19 and M 20–39	10–20	15	12
No 7.	M 20–39 and M 40–59	19–40	20	34
No 8.	M 40–59 and M 60+	19–40	20	27
The search for the optimized coefficients to be multiplied by the length of life expectancy				
No 9.	F 60+	1–2	1	1.4
No 10.	M 60+	1–2	1	1.2

We have applied the built-in ExtendSim Optimizer to minimize the mean percentage errors and mean absolute percentage errors (MAPE), as calculated between the historical values extracted from CSO databases [13] and the values obtained during the

simulation process. The differences were calculated every 1/100 years and subsequently integrated. This adopted configuration of optimization experiments is suitable, according to ExtendSim recommendations, for deterministic problems. We searched for the best maturation times between every pair of cohorts and the best lengths of life expectancy. For maturation times, we were looking for the best values given in years and for life expectancy for the coefficients to be multiplied by the historical value of the parameter.

5.2 Simulation: Optimization Results

The results of the optimization confirm the necessity of calibrating the maturation times. Figure 6 presents the MAPEs values, calculated between historical and simulation data describing the total WR population for when the model was loaded with the theoretical maturation times. Figure 7 shows MAPEs values calculated between historical and simulation data describing the total WR population, after the maturation times were modified according to the new values from the optimization experiment.

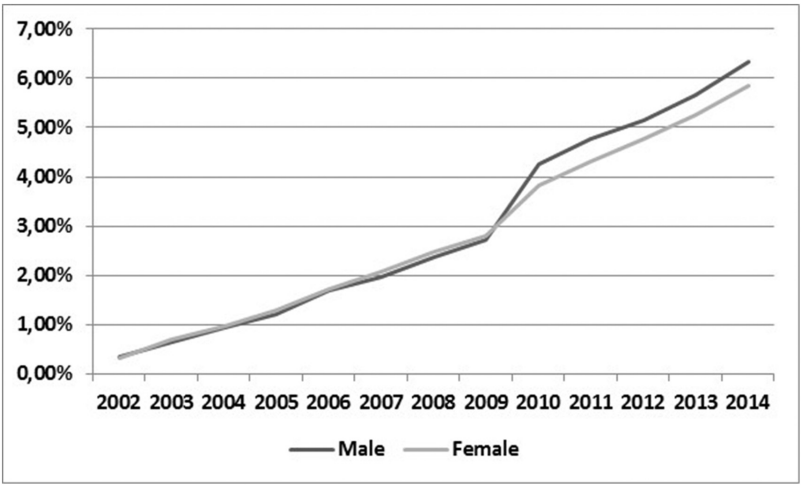


Fig. 6. MAPEs for female and male population between historical and simulation data. Maturation times and life expectancy are equal to the theoretical values.

The simulation experiment performed on the theoretical values of maturation times ended in 2014, with the MAPE values equal to 6.32% for male population and 5.85% for female population, as per Fig. 6. The experiment performed on the optimized maturation parameters produced much smaller MAPE values, equal to 1.99% for male population and 0.43% for female population, as per Fig. 7.

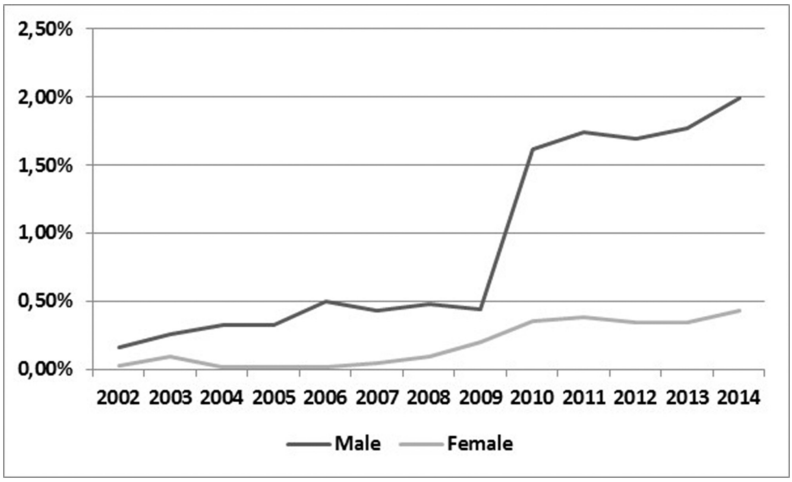


Fig. 7. MAPEs for female and male population between historical and simulation data. Maturation times and life expectancy are defined through the optimization process.

6 Model Testing and Calibration

The goal of the population submodel is to track the evolution of the WR population using aging chain cohorts. The model is run multiple times, and the primary function is to minimize the total differences between the number of males and females in particular cohorts. The output measures are the number of individuals in every cohort, as registered during the simulation at the end of the calendar year and the total population, for males and females respectively.

The simulation begins in 2002 and is run for 2002–2014 using historical parameters. This determines that the validation period starts in 2002 and ends in 2014. Figure 8 presents two age pyramids for 2014. The dark-coloured pyramid represents the distribution of the WR population based on historical data published by the CSO [13]. The light-coloured pyramid represents the simulation data.

Although there are differences in consecutive age cohorts between historical and simulation data when comparing the total number of individuals belonging to each cohort, the difference that relates to the total population is very small. The MAPEs calculated for the entire WR population in the particular years indicate that the simulation model provides, on average, acceptable results for the estimation of the WR population (Table 3). For particular age cohorts, in 2014, the MAPEs range from 0.18% to 12.73% (male population) and from 0.01% to 21.30% (female population).

The results demonstrate the usefulness of the SD approach in capturing the population evolution. It is not surprising that the differences between historical and simulation data begin to diverge (Fig. 7) only after a certain lapse of simulation time. Cohort blending is a slow process and, over a relatively short period of time, would be of only limited significance for slowly changing populations. However, it manifests more intensively as the time of simulation experiment increases.

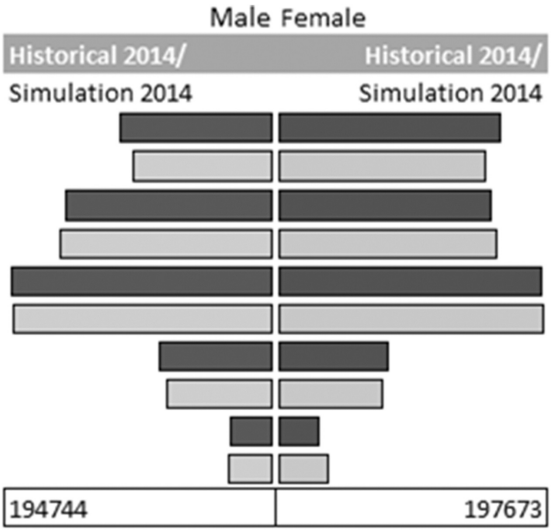


Fig. 8. Comparison of the pyramids of the WR population built from historical (dark colour) and simulation (light colour) data (Source: [26]).

Table 3. MAPEs calculated between historical and simulation data for total WR population in particular years.

Year	Male	Female
2002	0.16%	0.03%
2003	0.25%	0.10%
2004	0.33%	0.02%
2005	0.33%	0.02%
2006	0.50%	0.02%
2007	0.44%	0.05%
2008	0.48%	0.09%
2009	0.44%	0.20%
2010	1.61%	0.36%
2011	1.74%	0.39%
2012	1.69%	0.34%
2013	1.77%	0.34%
2014	1.99%	0.43%

(Source: [26])

7 Simulation Experiments

From among a range of probable scenarios of projection assumptions for population dynamics, as discussed by the Polish Government Population Council in 2014 [30], four scenarios were considered to be the most likely, but only one scenario was

officially recognized and published by the CSO. This scenario was adopted in our studies. We assumed that the development of the WR population will be affected by the demographic trends described in the official forecasts published by the CSO for 2014–2050. Below, we present the results of the simulation.

7.1 Simulation Scenario

The simulation begins in 2002 and runs through 2014 according to parameters calculated on the basis of historical values. However, maturation times and life expectancies were adjusted based on the output of the optimization algorithm. The following assumptions were included in the model for running the simulation beyond 2015.

- Fertility rates. It is assumed that fertility rates will incur a slight decline during the subsequent few years and, then, a gradual increase will be observed. By 2035, the increase in fertility rates values is expected to be approximately 14.7% (males) and 16.7% (females), as compared to 2014.
- Death rates. The death rates will constantly grow from 2015 to 2035 for all age groups, except for middle-aged cohorts (M 40–59 and F 40–59). For these cohorts, a slight decrease of death rates will first be observed. However, beyond 2030, the parameters will start to increase.
- Life expectancy. The difference between Poland and European countries will remain at a similar level during the next 20 years. This indicates that, in the year 2035, a woman aged 60 will live on average 27.75 years and a man 24.27 more years (24.39 years and 19.49 years in 2014, respectively).
- The difference between international and internal net migrations will decrease to almost zero. However, the total number of immigrants and emigrants will decrease by approximately 20%.

7.2 Simulation Results and Discussion

The simulation revealed many important demographic trends in the WR population, as per Fig. 9, and confirmed that the dominant trend for the WR, that is, population aging, is an irreversible phenomenon. The old-age rate, namely, the number of the oldest cohorts among the entire population, will increase from 22.99% in 2014 to 25.46% in 2035. The median age of population, that is, the age that half of the population has not yet reached and the other half has already lived, will, in 2035, be approximately 50 years. The next observation confirms another important trend related to the gender structure of the population. Simulation results show that female subpopulation will exceed the male subpopulation by more than 13%. The last observation demonstrates that, although the number of teenagers in 2035 will decrease, the total number of children 0–18 years old will be comparable to 2014.

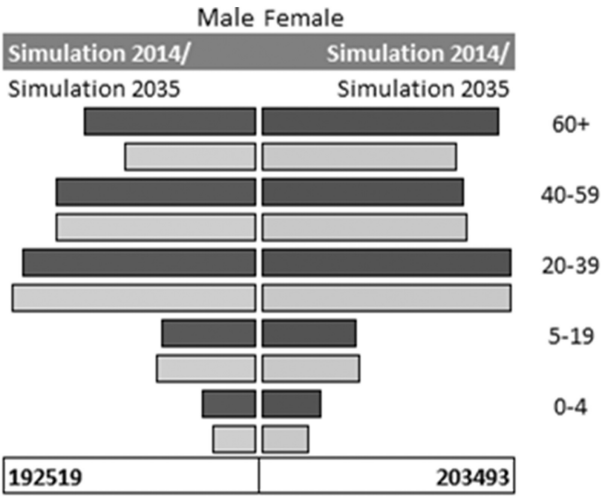


Fig. 9. Comparison of the age pyramids of the WR population built from the 2014 simulation (dark colour) and 2035 simulation (light colour) data (Source: [26]).

7.3 Future Work

This section indicates the direction of the next phase of the research, according to the goal of the project and resulting from the conclusions of this study. In the hybrid model, the states of the age-gender cohorts, as recorded continuously during the simulation in the SD module, will be directed to the DES module to generate the demand for healthcare services. Figures 10 and 11 present the preliminary findings for the F 60+ cohort.

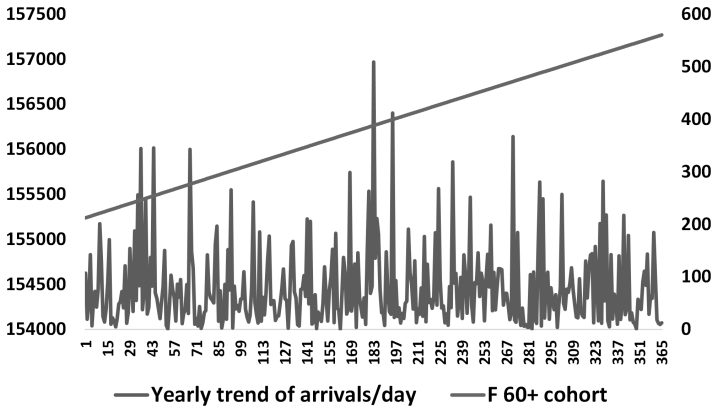


Fig. 10. Daily weekly trend of arrivals/day sampled from the F 60 + cohort (Source: [26]).

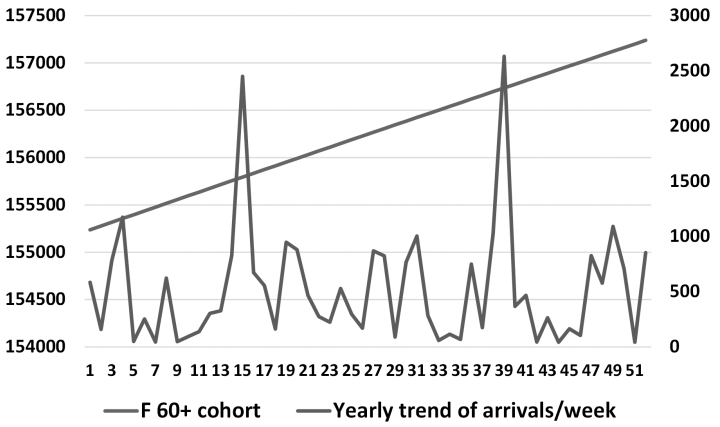


Fig. 11. Weekly trend of arrivals/day sampled from the F 60+ cohort (Source: [26]).

The stock level representing the total number of the F60+ population (the upper line) from 2014 to 2016 is converted to the flow of patients arriving to the healthcare system (the bottom line). The indistinguishable set of individuals was modelled by the discrete volume of needs-for-service expressed by females aged 60+ inhabiting the WR. The results of the simulation demonstrate the integration of two opposite perspectives: the projection of long-term population evolution based on aggregated data (the upper line) and the discrete input flow of individuals arriving to the healthcare system (the bottom line). The added value of this approach is the flexibility in the modelling of the arrival process. Figures 10 and 11 present the number of patients F60+ arriving to health units. Figure 10 was created based on daily sampling and Fig. 11 based on weekly sampling.

8 Conclusions

Demographic trends have a significant and stable effect on healthcare demand. The demand for healthcare services is strongly driven by uncertain factors, and some of these factors are closely related to on-going changes in age-gender population profiles. As such, valid forecasts may facilitate long-term planning strategies by health policy decision makers and enable health funds to accurately plan the future supply of services. This, in turn, might improve the equity of access to health services across the region and adjust the future regional budget to the changing needs of slowly but constantly evolving age-gender cohorts.

Because of the increasingly complex nature of problems faced when attempting to model healthcare systems, novel solutions are required. This research builds upon previous studies on using hybrid simulation to support healthcare decision making. We demonstrated the need of integrating the SD and DES methodologies to efficiently perform projections of population dynamics and forecast demand for healthcare services. SD is a well-known and often applied simulation technique to model demographic changes. The unique advantage of the SD model is its ability to capture the

dynamically evolving volatility of the human population. The drawback of this approach is that it produces perfectly mixed age/gender groups with indistinguishable individuals. This means that for individuals belonging to a particular cohort, the chances of becoming ill or dye are not related to their age. Moreover, the discrete event simulation easily captures individual choices made by patients. Individual attributes, such as age, place of residence, type of injury, and requested services, influence patients' decisions and consequently determine the utilization of healthcare resources.

Our goal was to construct a hybrid simulation model that would allow the linkage of demographic forecasts with a discrete model to predict future demand for healthcare services. In this study, we focused on the SD population submodel. The results demonstrate the usefulness of the approach in capturing the population evolution. The sampling procedure that reads and records values of stock levels enabled us to memorize individual attributes acquired by members of the population during their lifetimes. The results of the simulation experiments provide valuable insights into the dynamics of regional demographic trends and offer a well-defined starting point for future research in the health policy field.

The discussion presented in this paper is a first step toward more comprehensive studies, and potential research directions are as follows. The aging chain population model needs more extensive refinement and testing. Although the optimization algorithm ensured the correct simulation of the total population, the sizes of particular cohorts require additional calibration. The limitation of the model is that it assumes a given and stable level of the morbidity trend per age group and the absence of technological progress effect. We would also like to extend the model with some external/indirect incentives, such as economic growth, development of education or transportation infrastructure, and influence of the national pro-demography programme recently started by the Polish Government. The so-called family 500+ programme supports families having at least two children by granting monetary educational benefits. The programme is intended to increase fertility rates.

In addition, numerous technical problems need to be solved to better integrate the two modules driven by different simulation paradigms. For example, the single simulation run lasts around 25 min. This length of time is unacceptable when running a stochastic simulation that requires a number of independent replications.

Acknowledgements. This Project Was Financed by the Grant Simulation Modeling of the Demand for Healthcare Services from the National Science Centre, Poland, and Was Awarded based on the Decision 2015/17/B/HS4/00306.

References

1. Lisenkova, K., Mérette, M., Wright, R.: Population ageing and the labour market: modelling size and age-specific effects. Econ. Model. **35**, 981–989 (2013)
2. Tian, Y., Zhao, X.: Stochastic forecast of the financial sustainability of basic pension in China. Sustainability **8**, 46 (2016)

3. Lauf, S., Haase, D., Kleinschmit, B.: The effects of growth, shrinkage, population aging and preference shifts on urban development–a spatial scenario analysis of Berlin, Germany. Land Use Policy **52**, 240–254 (2016)
4. Ansah, J.P., Eberlein, R.L., Love, S.R., Bautista, M.A., Thompson, J.P., Malhotra, R., Matchar, D.B.: Implications of long-term care capacity response policies for an aging population: a simulation analysis. Health Policy **116**, 105–113 (2014)
5. Lutz, W., Sanderson, W., Scherbov, S.: The end of world population growth. Nature **412**, 543–545 (2001)
6. Lassila, J., Valkonen, T., Alho, J.M.: Demographic forecasts and fiscal policy rules. Int. J. Forecast. **30**, 1098–1109 (2014)
7. Davis, P., Lay-Yee, R., Pearson, J.: Using micro-simulation to create a synthesised data set and test policy options: the case of health service effects under demographic ageing. Health Policy **97**, 267–274 (2010)
8. Homer, J.B., Hirsch, G.B.: System dynamics modeling for public health: background and opportunities. Am. J. Public Health **96**, 452–458 (2006)
9. Barber, P., Lopez-Valcarcel, B.: Forecasting the need for medical specialists in Spain: application of a system dynamics model. Hum. Resour. Health **8**, 24 (2010)
10. Masnick, K., McDonnell, G.: A model linking clinical workforce skill mix planning to health and health care dynamics. Hum. Resour. Health **8**, 11 (2010)
11. Jun, J.B., Jacobson, S.H., Swisher, J.R.: Application of discrete-event simulation in health care clinics: a survey. J. Oper. Res. Soc. **50**, 109–123 (1999)
12. Mielczarek, B., Uziałko-Mydlikowska, J.: Application of computer simulation modeling in the health care sector: a survey. Simulation **88**, 197–216 (2012)
13. GUS, Główny Urząd Statystyczny. www.stat.gov.pl. Accessed Dec 2015
14. Sterman, J.D.: Business Dynamics. System Thinking and Modeling for a Complex World. McGraw-Hill Higher Education, Boston (2000)
15. Mustafee, N., Katsaliaki, K., Taylor, S.J.E.: Profiling literature in healthcare simulation. Simulation **86**, 543–558 (2010)
16. Testi, A., Tanfani, E., Torre, G.: A three-phase approach for operating theatre schedules. Health Care Manag. Sci. **10**, 72–163 (2007)
17. Sinreich, D., Marmor, Y.N.: Emergency department operations: the basis for developing a simulation model. IIE Trans. **37**, 233–245 (2005)
18. Hughes, G.R., Currie, C.S.M., Corbett, E.L.: Modeling tuberculosis in areas of high HIV prevalence. In: Perrone, L.F., Wieland, F.P., Liu, J., Lawson, B.G., Nicol, D.M., Fujimoto, R.M. (eds.) Proceedings of the 2006 Winter Simulation Conference, pp. 459–465. Institute of Electrical and Electronics Engineers, Inc., Piscataway (2006)
19. Kasaie, P., Kelton, W.D., Vaghefi, A., Naini, S.G.R.J.: Toward optimal resource-allocation for control of epidemics: an agent-based-simulation approach. In: Johansson, B., Jain, S., Montoya-Torres, J., Hugan, J., Yücesan, E. (eds.) Proceedings of the 2010 Winter Simulation Conference, pp. 2237–2248. Institute of Electrical and Electronics Engineers, Inc., Piscataway (2010)
20. Ashton, R., Hague, L., Brandreth, M., Worthington, D., Cropper, S.: A simulation-based study of a NHS walk-in centre. J. Oper. Res. Soc. **56**, 153–161 (2005)
21. Cardoso, T., Oliveira, M., Barbosa-Póvoa, A., Nickel, S.: Modeling the demand for long-term care services under uncertain information. Health Care Manag. Sci. **15**, 385–412 (2012)
22. Christie, P.M., Levary, R.R.: The use of simulation in planning the transportation of patients to hospitals following a disaster. J. Med. Syst. **22**, 289–300 (1998)
23. Han, L.D., Yuan, F., Shih-Miao, C., Hwang, H.: Global optimization of emergency evacuation assignments. Interfaces **36**, 502–513 (2006)

24. Caro, J.J., Guo, S., Ward, A., Shajil, C., Malik, F., Leyva, F.: Modelling the economic and health consequences of cardiac resynchronization therapy in the UK. Curr. Med. Res. Opin. **22**, 1171–1179 (2006)
25. Lane, D.C., Monefeldt, C., Rosenhead, J.V.: Looking in the wrong place for healthcare improvements: a system dynamics study of an accident and emergency department. J. Oper. Res. Soc. **51**, 518–531 (2000)
26. Mielczarek, B., Zabawa, J.: Modelling population growth, shrinkage and aging using a hybrid simulation approach: application to healthcare. In: Merkuryev, J., Oren, T., Obaidat, M.S. (eds.) Proceedings of the 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, SIMULTECH 2016, pp. 75–83. SciTePress (2016)
27. Eberlein, R.L., Thompson, J.P., Matchar, D.B.: Chronological aging in continuous time. In: Husemann, E., Lane, D. (eds.) Proceedings of the 30th International Conference of the System Dynamics Society (2011)
28. Mielczarek, B., Zabawa, J., Lubicz, M.: A system dynamics model to study the impact of an age pyramid on emergency demand. In: Obaidat, M.S., Kacprzyk, J., Oren, T. (eds.) Proceedings of the 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, pp. 879–888. SciTePress (2014)
29. Krahl, D.: ExtendSim advanced technology: discrete rate simulation. In: Rossetti, M.D., Hill, R.R., Johansson, B., Dunkin, A, Ingalls, R.G. (eds.) Proceedings of the 2009 Winter Simulation Conference, pp. 333–338. Institute of Electrical and Electronics Engineers, Inc., Piscataway (2009)
30. Waligórska, M., Kostrzewa, Z., Potyra, M., Rutkowska, L.: Population projection 2014–2050, CSO, Demographic Surveys and Labour Market Department (2014)

A Simulation-Based Dynamic Scheduling Method in Project Cost Estimation Process

Nobuaki Ishii[1(✉)], Yuichi Takano[2], and Masaaki Muraki[3]

[1] Faculty of Engineering, Kanagawa University, 3-27-1 Rokkakubashi, Kanagawa-ku, Yokohama 221-8686, Japan
n-ishii@kanagawa-u.ac.jp
[2] School of Network Information, Senshu University, 2-1-1 Higashimita, Tama-ku, Kawasaki 214-8580, Japan
ytakano@isc.senshu-u.ac.jp
[3] Graduate School of Decision Science and Technology, Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo 152-8550, Japan
m.muraki85ll@gmail.com

Abstract. Since project price is determined before the start of a project, project cost estimation is a critical work for the EPC (Engineering-Procurement-Construction) contractor in accepting profitable projects in competitive bidding situations. The contractor should devote significant time and resources to accurate cost estimation of project orders from clients. However, it is impossible for any contractor to devote significant time and resources to all the orders because such resources are usually limited. For this reason, the contractor must dynamically decide bid or no-bid on the orders at each order arrival, and allocate the limited resources to the chosen orders. In this paper, we develop a simulation model of the project cost estimation process by reference to a generic model of dynamic scheduling for the state-dependent work. Then we devise a simulation-based method for dynamic scheduling in the project cost estimation process by using the model to maximize the contractor's profits. The method dynamically selects orders and allocates the limited resources to them, on the basis of the contractor's resource utilization, and the expected profit from the order. The effectiveness of our method is demonstrated through simulation experiments.

Keywords: Competitive bidding · Discrete event simulation · Project management · Project selection · Resource allocation

1 Introduction

Project cost estimation is a critical work for the contractor in accepting profitable EPC (Engineering-Procurement-Construction) projects in competitive bidding situations. In EPC projects, the contractor delivers unique facilities, such as process plants, structures, information systems, and so on, based on the client's requirements for a limited period of time under a lump sum turnkey basis [1]. Since any EPC project includes unique and non-repetitive activities, many uncertainties exist in the project execution process. Furthermore, since the project price is fixed before the start of the project, the contractor often faces eventual loss in EPC projects. Thus, any contractor

M.S. Obaidat et al. (eds.), *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing 676,
https://doi.org/10.1007/978-3-319-69832-8_15

who seeks to increase profits and reduce the possibility of realizing a loss, i.e., deficit risk, due to cost estimation error, should precisely estimate the project cost in order to determine the bidding price [2].

Since the quality and quantity of the data available for cost estimation determine the accuracy of estimated cost, a large amount of high-quality data is required to improve accuracy. In process plant engineering, for example, AACE International [3] has studied the data and methods that are required to attain the target accuracy of project cost estimation. In fact, various cost estimation methods, such as parametric, analogy, and engineering, are used in practice [4]; however, in any method, higher accuracy needs more data and, accordingly, requires more engineering Man-Hours (hereafter referred to as MH) to acquire and analyze the data for cost estimation.

Thus, experienced and skilled human resources who can acquire data and create project plans are required for accurate cost estimation. However, those resources are limited for any contractor; furthermore, once the orders are successfully accepted, the corresponding project execution will also need considerable human resources. For these reasons, the contractor should realize appropriate allocation of MH for cost estimation to each order to maximize the total expected profit under the constraint of total MH [2]. The contractor should also consider the deficit risk, due to cost estimation error. This is because just a few deficit orders, which produce an eventual loss due to cost estimation error, would result in a significant reduction of contractor's profits when the number of accepted orders is small.

This paper examines the cost estimation process of EPC projects in dynamic order arrival situations. Then, we develop a simulation-based method that dynamically selects orders and allocates MH for cost estimation to each selected order to maximize the expected profits. For this purpose, we begin by building a simulation model of the cost estimation process with reference to a generic model of the dynamic scheduling for the state-dependent work and previous studies by Ishii et al. [5, 6]. In the state-dependent work, work process, resources and time can be changed according to the work situation. Sales, research and development, software testing, education, and training, and so on are the typical examples of the state-dependent work.

In our simulation model, the cost estimation process is divided into four activities, i.e., order selection, Class 4 estimate, Class 3 estimate, and Class 2 estimate, based on the AACE cost estimate classification system [3] that indicates the methods, data, and accuracy of cost estimation in each class. We next establish the order selection rules for deciding bid or no-bid on arrived orders based on the threshold function of MH utilization with respect to the expected profit of orders. This threshold function is created through simulation experiments using our simulation model of the cost estimation process. We finally analyze the effectiveness of our simulation-based method through numerical examples.

2 Related Work

A variety of studies have been conducted on project cost estimation from the viewpoints of competitive bidding, cost estimation accuracy, resource allocation, order selection, and so on.

For example, a number of researchers have conducted the studies on the competitive bidding strategy [7] since Friedman [8] presented a method to determine an optimal bidding price based on the distribution of the ratio of the bidding price to cost estimate. However, little attention has been paid to profit volatility risk arising from cost estimation error, which cannot be ignored in EPC projects. When, for instance, the number of accepted orders is limited, the realized total profit from the projects might be sharply lower than expected because the profit is significantly affected by a few deficit orders. Accordingly, the accuracy of cost estimation should be considered in the EPC projects.

Oberlender and Trost [9] studied determinants of cost estimation accuracy and developed a system for predicting accuracy. Bertisen and Davis [10] analyzed the costs of 63 projects and evaluated the accuracy of estimated costs statistically. Jørgensen et al. [11] studied the relationship between project size and cost estimation accuracy. Uzzafer [12] proposed a contingency estimation model in consideration of the distribution of estimated cost and the risk of software projects to estimate contingency resources.

In addition, Humphreys [13], Towler and Sinnott [14], and AACE International [3] demonstrated the relationship in cost estimation accuracy and the method and data used for cost estimation in the field of process plant engineering projects. Furthermore, they suggested that cost estimation accuracy is positively correlated with the volume of MH for cost estimation.

Regarding resource allocation, several papers have analyzed the problem of allocating scarce resources in competitive bidding (see Rothkopf and Harstad [15] for detailed references). Among them, Kortanek et al. [16] considered sequential bidding models where the obtained contracts require the use of restricted resources, such as production capacity, at the time of actual production. In addition, several studies that focus on the volume of MH for cost estimation and cost estimation accuracy have been conducted. For example, Ishii et al. [17, 18] developed an algorithm that determines the bidding prices under limited MH for cost estimation. Their algorithm allocates MH to the orders so as to maximize expected profits based on the cost estimation accuracy determined by allocated MH. In addition, Takano et al. [19] developed a stochastic dynamic programming model for establishing an optimal sequential bidding strategy in a competitive bidding situation. Their model determines the optimal markup in consideration of the effect of inaccurate cost estimates. Furthermore, Takano et al. [20] developed a multi-period resource allocation method for estimating project costs in a sequential competitive bidding situation. Their method allocates resources for cost estimation by solving a mixed integer programming problem that is formulated by making a piecewise liner approximation of the expected profit functions.

Regarding the order selection in the cost estimation process, Shafahi and Haghani [21] propose an optimization model that combines project selection decisions and markup selection decisions in consideration of eminence and previous works as the non-monetary evaluation criterion used by owners for evaluating bids.

Based on the above literature review, we can say that most studies have paid little attention to the project cost estimation process in practical situations. More specifically, the contractor needs to allocate MH for cost estimation dynamically to each arrived orders with different attributes in practice. To the best of our knowledge, however, none of the existing studies except Ishii et al. [5, 6] have investigated the project cost

estimation process in dynamic order arrival situations. In light of these facts, this paper develops a simulation-based dynamic scheduling method for selecting orders and determining MH allocation dynamically in consideration of the contractor's available MH and the orders' profitability.

3 A Generic Scheduling Problem in State-Dependent Work

In general, scheduling is a decision problem that develops a plan with reference to the sequence of required works and time allocated for each item or activity necessary to complete them under the constraints of process conditions, the amount of resources, and so on. The goal of scheduling is to optimize one or more objectives, such as cost, makespan, and so on. Although there are many scheduling problems in practice, the production scheduling problems [22, 23] have been well studied. The production scheduling problem is divided into several categories, based on the production system, characteristics of data used, assumption of order arrivals, and so on. For example, in the typical scheduling research, the scheduling problem has been addressed as static scheduling problems and dynamic scheduling problems. The static problem consists of a pre-defined set of orders. In contrast, the dynamic problem deals with continuously arriving orders. However, in any case, the deliverables are predetermined, and orders are scheduled so as to minimize the makespan or cost in the production scheduling. No valuable deliverables can be obtained if the activities are eventually terminated in the mid-process of the schedule.

In contrast, the project cost estimation process is classified as state-dependent work that creates non-physical products, unlike the case of production. In the state-dependent work, the work process, resources and time, and therefore the value of deliverables from the work, can be changed according to the work situation. Sales, software testing, research and development, education, and training, and so on are the typical examples of state-dependent work.

The state-dependent work has the following characteristics,

- Work activities can be terminated in the mid-process,
- Deliverables gained through the work have some value according to the amount of resources and time invested, no matter which work activities are terminated in the mid-process,
- The value of the deliverables is evaluated based on the state when the work was terminated or completed.

The project cost estimation process creates design documents, project cost, schedule, etc., which have some value according to the amount of MH used if the estimation activities are not completed. Accordingly, we can state that the project cost estimation process has the above characteristics, and thus it can be classified as state-dependent work.

Since the above characteristics are different from those of the ordinary production process from the viewpoints of the scheduling problem as shown in Table 1, different approaches should be taken in the scheduling of state-dependent work.

Table 1. Scheduling problems in state-dependent work and production process.

	State-dependent work	Production
Evaluation	Expected profits, Amount of accepted orders, Customer satisfaction, etc.	Cost, Makespan
Process	Can change or skip by condition	Cannot change. Entire process must be completed
Value of deliverables	Vary according to resource used	Not vary

3.1 A Generic Model of Dynamic Scheduling for State-Dependent Work

Based on the characteristics of the state-dependent work, we developed a generic model of dynamic scheduling for the state-dependent work as shown in Fig. 1 [24].

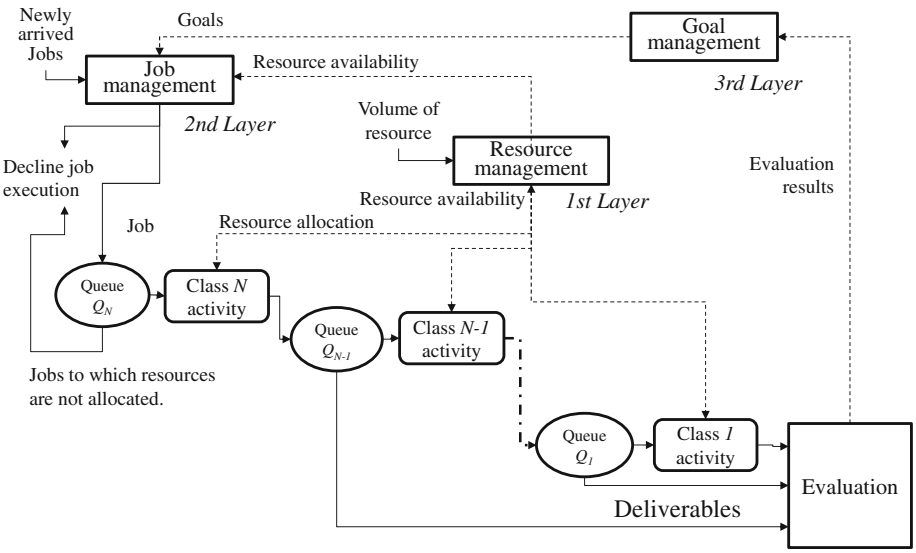


Fig. 1. Generic model of dynamic scheduling for state-dependent work.

In the model, we suppose that the work can be divided into N classes of activities, i.e., Class N, Class $N - 1$, …, and Class 1 activity, and are carried out from Class N to Class 1, sequentially. Jobs arrive randomly and are first filed in the queue for the Class N activity and wait to be assigned resources to carry out the Class N activity by the mechanism of resource management. If any resource is not assigned to the job by the due date, then no activity is carried out due to lack of resources. If the resources are assigned to the job, the activity is performed by creating the value of the Class N activity. The job is then filed in the queue of the Class $N - 1$ activity and waits for

resource assignment for the Class $N - 1$ activity. If the resources are not further assigned to the job by the due date, the deliverables are evaluated based on the results by the activity of Class N. By contrast, if the resources are assigned to the job waiting in the queue of the Class $N - 1$ activity, the activity is done, and then filed in the queue of the Class $N - 2$ activity. In our model, the same decision is made for the jobs in the queue of each class until they complete the Class 1 activity or terminate the work on the way to Class 1 activity.

The value of the deliverables in the state-dependent work gradually improves through the activities in each class because the scope of the activity is wider and more detailed according to the progress of activities. However, the work can be terminated on the way to Class 1 activity; in such a case, the value of deliverables created through the work is determined by the final class carried out. Then, the deliverables are evaluated, and the goals of the project cost estimation process are modified according to the work results.

In addition, the generic model of dynamic scheduling in the state-dependent work shown in Fig. 1 assumes to manage works with a three-layer management structure as follows:

- 1st Layer (Resource management): allocates resources required to carry out the activities of the job waiting in a queue file within the available resource,
- 2nd Layer (Job management): decides whether to decline the activities on the newly arrived job based on the goals set in goal management,
- 3rd Layer (Goal management): sets and changes goals in an appropriate time based on the results of state-dependent work.

3.2 Scheduling Method

A variety of methods have been developed for scheduling systems in both academia and industry. In this paper, we use the simulation-based method, which can adapt to the dynamic job arrival situation in the state-dependent work.

Figure 2 shows the basic structure of the simulation-based dynamic scheduling consisting of three modules, i.e. simulation, evaluation, rule management [24]. By using the model of dynamic scheduling, the simulation module simulates the state-dependent work under several scenarios, including job arrival conditions, simulation rules, and operating conditions of the actual work environments, such as resource conditions, and current working jobs. The results of simulations are evaluated in the evaluation module, then the simulation rules are modified by the rule management module, if necessary. The simulation-based method searches the best simulation rules throughout the mechanism. Then, the simulation rules that perform best under the simulation environments are used for controlling jobs in the actual work environments.

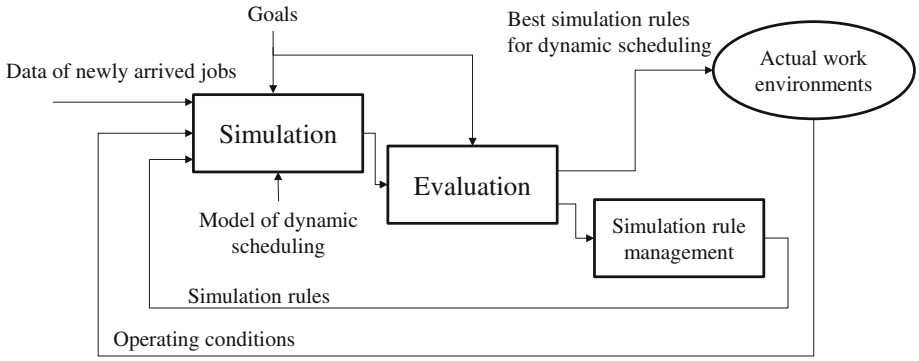


Fig. 2. Basic structure of simulation-based dynamic scheduling.

4 A Model of Project Cost Estimation Process

As is the case with the state-dependent work, the project cost estimation process can be divided into a series of activities that starts with the arrival of bid invitations and closes by the date of bidding, i.e. due date. A variety of orders arrive, and the cost of projects is estimated through the project cost estimation process. We decide the accuracy of cost estimation by allocating MH to the cost estimation activities of newly arrived orders in consideration of the MH availability, expected profits, competitive bidding situations, and so on. When the available MH is not enough to estimate cost accurately, we must allocate less MH, thereby reducing expected profit due to inaccurate cost estimation or no-bid on the order.

Based on the above observations, we propose a model of the project cost estimation process as shown in Fig. 3 [5, 6] by reference to the generic model for dynamic scheduling in the state-dependent work. In the model, we assume that the cost is estimated through three classes: Class 4, Class 3, and Class 2 estimate. Each class needs MH and a period of time for cost estimation, and the accuracy of estimated cost increases through the cost estimation activities in each estimate class. The cost estimate classification matrix [3] can be used to set the cost estimation accuracy in each class.

The model of the project cost estimation process shown in Fig. 3 assumes to manage the cost estimation process based on a three-layer management structure, which modifies the generic model as follows:

- 1st Layer (MH allocation): allocates required MH to the orders waiting for cost estimation,
- 2nd Layer (Order selection): decides whether to bid on the newly arrived order,
- 3rd Layer (Goal setting): evaluates the results of bidding, and modifies goals, if necessary.

In the model, the order selection module decides whether to bid on the newly arrived order from the viewpoint of the volume of orders to be accepted, the expected profits, MH availability for cost estimation, and so on. The selected order is first filed in the queue for the Class 4 estimate and waits to be assigned the MH for cost estimation

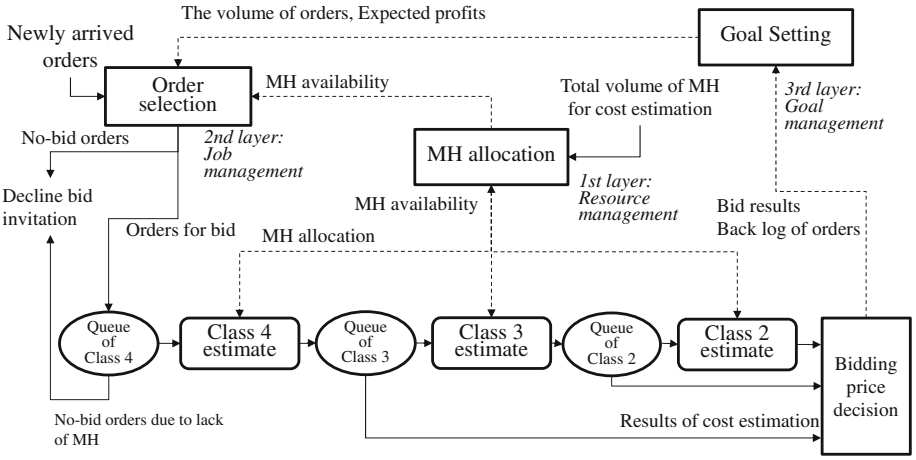


Fig. 3. Model of project cost estimation process.

by the mechanism of MH allocation for cost estimation. If any MH is not assigned to the order by the bidding date, the contractor does not bid for it due to lack of MH. If MH is assigned to the order, its project cost is estimated with the accuracy of the Class 4 estimate. This order is then filed in the queue of the Class 3 estimate and waits for MH assignment for the Class 3 estimate. If the MH is not further assigned to the order by the bidding date, the contractor decides the bidding price based on the accuracy of the Class 4 estimate. By contrast, if the MH is assigned to the order, it is estimated with the accuracy of the Class 3 estimate, and then it is filed in the queue of the Class 2 estimate. The same decision is made for the orders in the queue of the Class 2 estimate.

5 Simulation-Based Method

This section shows a simulation-based method for dynamic scheduling in the project cost estimation process based on the scheduling method and the model of the project cost estimation process described in Sects. 3 and 4, respectively. Our simulation-based method is developed based on the following assumptions:

Assumptions:

1. Orders for cost estimation arrive randomly;
2. Expected profit, required MH and periods for cost estimation of each estimate class are predetermined;
3. Probability of a successful bid of each order is predetermined.

Since EPC contractors can collect their own data on past projects and market situations, assumptions 2 and 3 are appropriate.

5.1 Order Selection Mechanism

The simulation-based method for dynamic scheduling uses two mechanisms, i.e. order selection and MH allocation for cost estimation, to simulate the project cost estimation process. The order selection mechanism selects orders for cost estimation based on order selection rules. The MH allocation mechanism assigns the MH for cost estimation to each selected order, so as to maximize the expected profits from orders.

Order Selection Method
The order selection method [6] is based on the financial evaluation criteria and consists of the following two steps:

Step 1: Calculate the expected profit per MH for cost estimation of the new arrival order i as follows:

$$EPPC_i = EP_i / EM_i \tag{1}$$

where $EPPC_i$ is the expected profit per MH for cost estimation of order i, EP_i is the expected profit of order i, and EM_i is the volume of MH required to estimate the cost of order i. In this paper, $EPPC_i$ is calculated based on the Class 2 estimate in AACE cost estimate class [3].

Step 2: Make the bid/no-bid decision on the new arrival order by considering $EPPC_i$ of the order and the contractor's MHU, which is the volume of MH being utilized for cost estimation at the time of new order arrival. For this purpose, we use a threshold function $MHU_{up}(EPPC_i)$, which indicates the upper limit of MHU in selecting order i for cost estimation, as follows:

- The contractor selects the new arrival order i for cost estimation if MHU is lower than $MHU_{up}(EPPC_i)$;
- Otherwise, the contractor decides not to bid on the order.

The contractor can expect higher profits from the order by estimating its project cost in a higher cost estimate class. However, more MH is required for estimating cost in a higher cost estimate class. In the above steps, the new arrival orders with low expected profits are not selected for cost estimation when a large volume of MH is being utilized for cost estimation. Thus, this method eliminates a possible shortage of MH for cost estimation and, accordingly, allows the contractor to focus on estimating cost of profitable orders. In other words, our order selection method works to maintain the balance between order's profitability and contractor's MH utilization so that the contractor's expected profits are maximized in dynamic order arrival situations.

Determination of Threshold Function
In our model, orders with different attributes arrive randomly in a project cost estimation process. Thus the MH utilization changes dynamically and unpredictably. Consequently, it is very difficult to find a threshold function $MHU_{up}(EPPC_i)$ for maximizing contractor's expected profits.

In view of these observations, we develop a method that searches the threshold function [6] by using the simulation model shown in Fig. 3. This method searches three threshold points, P1(E_1, N_1), P2(E_2, N_2) and P3(E_3, N_3), sequentially by applying them

in the order selection mechanism. As shown in Fig. 4, the no-bid area is expressed as follows:

$$\cup_{k=1}^{3}\{(EPPC, MHU)|EPPC \leq E_k, MHU \geq N_k\} \quad (2)$$

The threshold function $MHU_{up}(EPPC_i)$ marks the boundary between the no-bid area and cost estimation area. The procedure of the simulation-based method is described as follows:

Step 1: Set all the threshold points to (0, 0).
Step 2: Search P2(E_2, N_2) that maximizes the expected profit by running a simulation under the current conditions, i.e., order arrival interval, cost estimation period and required MH in each class of cost estimate, and expected profit of each order.
Step 3: Search P1(E_1, N_1) that maximizes the expected profit by running a simulation, where P2(E_2, N_2) is fixed to the value searched in Step 2.
Step 4: Search P3(E_3, N_3) that maximizes the expected profit by running a simulation, where P1(E_1, N_1) and P2(E_2, N_2) are fixed to the values searched in Steps 2 and 3.
Step 5: Define $MHU_{up}(EPPC_i)$ as the boundary formed by P1(E_1, N_1), P2 (E_2, N_2) and P3(E_3, N_3) as shown in Fig. 4.

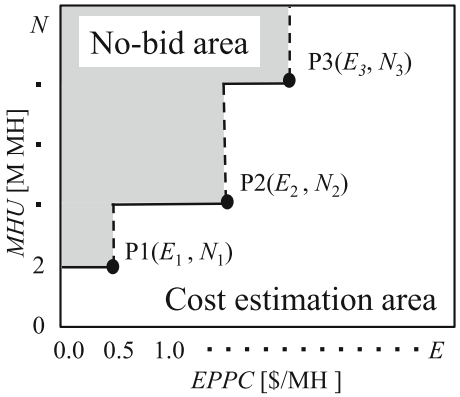


Fig. 4. Area of bid/no-bid decision [6].

5.2 Allocation of MH for Cost Estimation

For the allocation of MH for cost estimation, we shall use a dispatching approach, as is the case with the dynamic scheduling problem in production processes [23].

Specifically, when MH is released from cost estimation of an order, this approach selects an order based on the dispatching rules, which prioritize orders in the queue of each estimate class. The selected order is subsequently assigned the required MH for its estimate class. If the required MH is more than the MH available, the selected order waits in the queue until the required MH is released.

One can use well-known dispatching rules for the allocation of MH, such as FIFO, SPT, and EDD [23]; however, dedicated rules for a project cost estimation process can also be designed. In addition, the advanced dispatching rule approach, which changes dispatching rules dynamically according to the conditions of MH availability, conditions of orders waiting for cost estimation activities in the queues, and so on, would be effective as is the case with production systems [25].

6 Numerical Examples

This section evaluates the effectiveness of our simulation-based method for dynamic scheduling based on the data and conditions used in the paper by Ishii et al. [6]. For the simulation experiments, we use a general-purpose simulation system AweSim! [26].

6.1 Design of Simulation Experiments

To determine the threshold function $MHU_{up}(EPPC_i)$, we use the scenario selection system developed by Nelson et al. [27]. This system statistically compares the results of simulation runs and chooses sequentially the best threshold points P2(E_2, N_2), P3(E_3, N_3), P1(E_1, N_1) from candidate points given by us. The volume of MH is set to 16,000 MH per period, i.e., 16 [M MH], and the simulation period is set to 1200.

It is supposed that there are orders of the three sizes, i.e., Small, Medium, Large, in our simulation experiments. For these orders, we consider three cases—Case 1, Case 2, and Case 3—that have different expected profit of the Class 3 estimate, as shown in Table 2. In addition, we consider three sub-cases—Case A, Case B, and Case C—based on the order arrival intervals defined by the triangular distribution, as shown in Table 3. In what follows, Case 1.A means that both Case 1 and Case A are considered. Table 4 shows parameters of triangular distribution that represents the probability of order acceptance in each order size. It follows that by bidding for an order, the expected profit shown in Table 2 is gained with the associated probability of order acceptance. Table 5 shows cost estimation conditions of each cost estimate class, i.e., total periods available for cost estimation (due date for bidding), required periods and MH for cost estimation.

Table 2. Expected profit of orders (all cases) [MM$] [6].

		Order size		
		Small	Medium	Large
Case 1	Class 4	0.5	1	1.5
	Class 3	5	10	15
	Class 2	20	40	60
Case 2	Class 4	0.5	1	1.5
	Class 3	10	20	30
	Class 2	20	40	60
Case 3	Class 4	0.5	1	1.5
	Class 3	15	30	45
	Class 2	20	40	60

Table 3. Order arrival interval [Orders/Period] [6].

	Parameters of triangular distribution	Order size		
		Small	Medium	Large
Case A	Min.	1.05	2.70	3.15
	Mode	1.50	3.00	4.50
	Max.	1.95	3.90	5.85
Case B	Min.	0.84	1.68	2.52
	Mode	1.20	2.40	3.60
	Max.	1.56	3.12	4.68
Case C	Min.	0.70	1.40	2.10
	Mode	1.00	2.00	3.00
	Max.	1.30	2.60	3.90

Table 4. Probability of order acceptance (all cases) [6].

		Order size		
		Small	Medium	Large
Parameters of triangular distribution	Min.	0.05	0.05	0.05
	Mode	0.20	0.30	0.40
	Max.	0.90	0.90	0.90

Table 5. Cost estimation conditions (all cases) [6].

		Order size		
		Small	Medium	Large
Total periods available for cost estimation		8	8	8
Periods for cost estimation	Class 4	1	1	1
	Class 3	2	2	2
	Class 2	3	3	3
MH for cost estimation [M MH]	Class 4	1	2	3
	Class 3	2	3	4
	Class 2	3	4	6

Our simulation experiments evaluated each case by using the following order selection rules and dispatching rules.

Order Selection Rule

The following two order selection rules, i.e. No selection and MHU basis are tested in the experiments.

- No selection: All the arrived orders are selected for cost estimation.
- MHU basis: Orders are selected for cost estimation by the order selection mechanism described in Sect. 5.

Dispatching Rule for Allocating MH for Cost Estimation
The following three dispatching rules, i.e. FIFO, HEPF, and HACF are tested in the experiments.

- FIFO: Orders are selected on a first-in first-out basis.
- HEPF: Order of the largest increment of *EPPC* is selected first.
- HACF: Order of the highest acceptance probability first.

6.2 Results of Simulation Experiments

Figures 5, 6, and 7 depict the threshold function $MHU_{up}(EPPC_i)$ together with the threshold points P1(E_1, N_1), P2(E_2, N_2) and P3(E_3, N_3) determined by our simulation-based method for Cases 1.A, 1.B, and 1.C, respectively. For example, when 10 MH is being utilized by the contractor, the arrived order of 0.7 EPPC is selected for the cost estimation in Case 1.A, however, it is not selected for the cost estimation in Cases 1.B and 1.C.

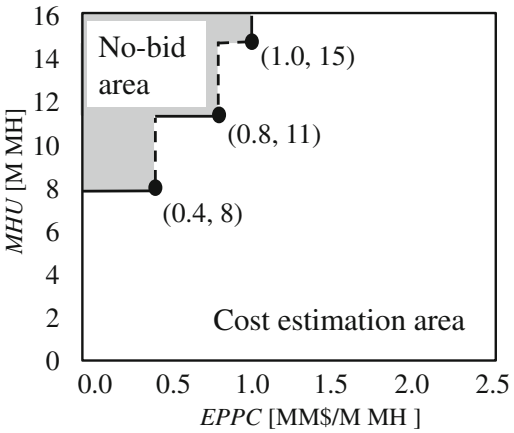


Fig. 5. Area of bid/no-bid decision in Case 1.A [6].

We can see in the figures that the no-bid area becomes wider according to the increase of the number of arrived orders in the cost estimation process. Indeed, Case 1.C, where orders arrive most frequently among all cases, has the widest no-bid area. It is also found from the figures that in making bid/no-bid decisions, Case 1.C puts a high priority on the order's expected profit, whereas Case 1.A takes into account both the order's expected profit and the contractor's MH utilization. This implies that contractors should pay attention to its MH utilization for cost estimation especially when the number of arrival orders is limited.

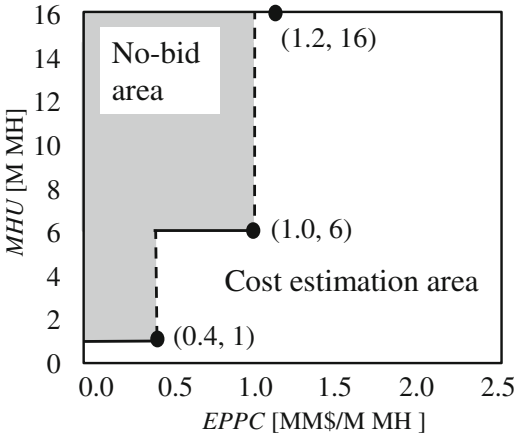


Fig. 6. Area of bid/no-bid decision in Case 1.B [6].

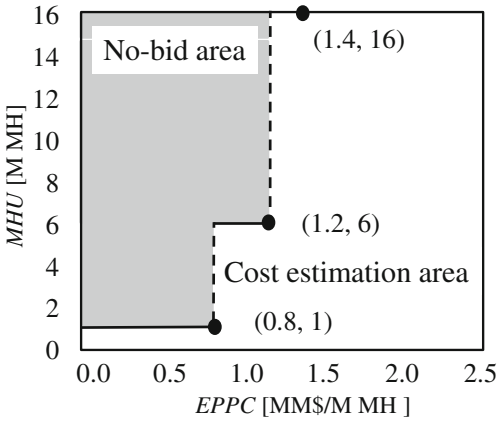
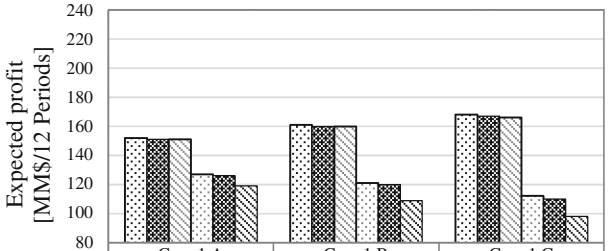


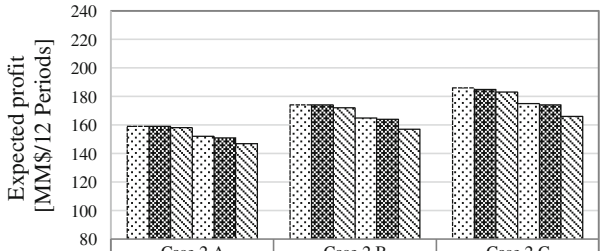
Fig. 7. Area of bid/no-bid decision in Case 1.C [6].

Figures 8, 9, and 10 show the expected profits of each combination of order selection rules and MH allocation rules. Regarding the order selection rule, the MHU basis rule gains larger expected profits than the no selection rule does. For example, in Case 1.C, the expected profit by MHU basis HEPF is 168 [MM$], and that by no selection HEPF is 112 [MM$]. In addition, the improvement in the expected profits by the MHU basis rule increases according to the increase of the number of arrived orders in the project cost estimation process. In fact, the ratio of improvement in the expected profits by MHU basis HEPF is about 22%, 34%, and 50% against the no section rule, in Cases 1.A, 1.B and 1.C, respectively.



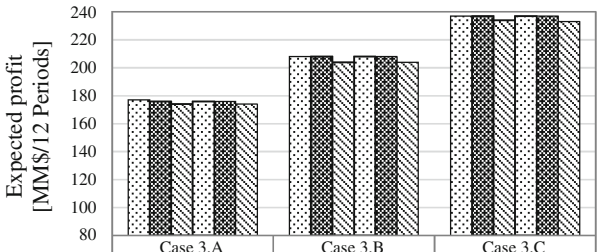
	Case 1.A	Case 1.B	Case 1.C
MHU basis HEPF	152	161	168
MHU basisi HACF	151	160	167
MHU basisi FIFO	151	160	166
No slection HEPF	127	121	112
No slection HACF	126	120	110
No selection FIFO	119	109	98

Fig. 8. Expected profits in Case 1.



	Case 2.A	Case 2.B	Case 2.C
MHU basis HEPF	159	174	186
MHU basisi HACF	159	174	185
MHU basisi FIFO	158	172	183
No slection HEPF	152	165	175
No slection HACF	151	164	174
No selection FIFO	147	157	166

Fig. 9. Expected profits in Case 2.



	Case 3.A	Case 3.B	Case 3.C
MHU basis HEPF	177	208	237
MHU basisi HACF	176	208	237
MHU basisi FIFO	174	204	234
No slection HEPF	176	208	237
No slection HACF	176	208	237
No selection FIFO	174	204	233

Fig. 10. Expected profits in Case 3.

On the other hand, as shown in Fig. 10, the effects of the MHU basis rule on the expected profits are very small in Case 3. The main reason is that in Case 3, the expected profits of the Class 3 estimate are close to those of the Class 2 estimate as shown in Table 2. No selection rules allocate MH for cost estimation evenly to all the orders and, accordingly, increase the number of Class 3 estimates. As a result, this rule works well only in Case 3. By contrast, the MHU basis rules make bid/no-bid decisions based on the threshold functions as shown in Figs. 5, 6, 7, and thus, they work effectively in all cases.

Regarding the dispatching rules for allocating MH, HEPF and HACF rules perform slightly better than FIFO rules. However, they make no significant difference in the expected profits, especially when the MHU basis rule is used for order selection. The MHU basis rule removes orders having low expected profit at the gate of the cost estimation process. Thus we guess that the dispatching rules that allocate MH for cost estimation to the orders based on the expected profit, like HEPF and HACF rules, cannot make a significant difference.

Tables 6, 7, and 8 show the ratio of cost estimate class determined by the HEPF rule. The MHU basis rule makes many Class 2 estimates compared with the no selection rule in Cases 1 and 2. Additionally, we observe that the number of no-bid orders is also large in the MHU basis rule. For example, the MHU basis rule makes no-bid decisions on 38.7% of arrived orders in Case 1.A as shown in Table 6. In the case of the MHU basis rule, the ratio of no-bid orders increases according to the increase of number of arrived orders in the project cost estimation process. Namely, the ratio of no-bid orders increases as 38.7%, 50.4%, and 62.0% according to the increase of the number of arrived orders in Case 1.A, Case 1.B, and Case 1.C. This maintains the number of the Class 2 and Class 3 estimates, which bring more expected profits than the Class 4 estimate.

Table 6. Ratio of cost estimate class in Case 1 HEPF rule (MHU: MHU basis, No: No selection) [%] [6].

	Case 1.A		Case 1.B		Case 1.C	
	MHU	No	MHU	No	MHU	No
No-bid	38.7	0.0	50.4	0.0	62.0	0.0
Class 4	0.0	0.0	0.0	0.1	0.0	0.6
Class 3	7.6	50.1	8.5	71.9	6.2	87.0
Class 2	53.7	49.9	41.2	28.1	31.8	12.3

In our simulation, the average number of arrived orders is 1465, 1827, and 2195 in Cases A, B, and C, respectively. In Case 3, however, the ratio of cost estimate class provided by the MHU basis rules is very similar to that provided by no selection rules as shown in Table 8. Since the expected profits per MH of the Class 3 estimate is higher than that of the Class 2 estimate in Case 3, the MHU basis rule focuses MH for cost estimation on the Class 3 estimates.

Table 7. Ratio of cost estimate class in Case 2 HEPF rule (MHU: MHU basis, No: No selection) [%] [6].

	Case 2.A		Case 2.B		Case 2.C	
	MHU	No	MHU	No	MHU	No
No-bid	31.8	0.0	32.7	0.0	47.4	0.0
Class 4	0.0	0.0	0.0	0.1	0.0	0.6
Class 3	13.3	50.1	28.4	71.9	21.5	87.0
Class 2	54.9	49.9	38.9	28.1	31.1	12.3

Table 8. Ratio of cost estimate class in Case 3 HEPF rule (MHU: MHU basis, No: No selection) [%] [6].

	Case 3.A		Case 3.B		Case 3.C	
	MHU	No	MHU	No	MHU	No
No-bid	0.6	0.0	0.7	0.0	0.9	0.0
Class 4	0.0	0.0	0.1	0.1	0.6	0.6
Class 3	49.4	50.1	71.3	71.9	85.5	87.0
Class 2	50.0	49.9	28.0	28.1	13.0	12.3

These observations confirm that our order selection mechanism in the simulation-based method works well to allocate MH for cost estimation appropriately so that the expected profits from orders are maximized in the dynamic order arrival situations.

7 Conclusion

In this paper, we explore the project cost estimation process of EPC projects in dynamic order arrival situations. The simulation model of the cost estimation process is made by reference to the generic model for dynamic scheduling in the state-dependent work, and then a simulation-based method is developed for dynamic scheduling in the project cost estimation process. It selects orders for cost estimation based on order selection rules and allocates MH for cost estimation to each selected order to maximize the expected profits from orders. In this method, order selection rules decide bid or no-bid on arrived orders by using the threshold function $MHU_{up}(EPPC_i)$. This function is determined through simulation experiments using the model of the project cost estimation process. In addition, dispatching rules prioritize orders in the queue of each estimate class to assign MH effectively. We analyze the effectiveness of our simulation-based method in terms of the expected profit through numerical examples.

The following conclusions can be drawn from the analysis of the numerical examples:

- Our simulation-based method works well to allocate MH for cost estimation appropriately so that the expected profits from orders are maximized in the dynamic order arrival situations.

- HEPF, HACF, and FIFO rules, which are used to dispatch orders waiting for cost estimation, make no significant difference in the expected profits, especially when the MHU basis rule is used for order selection.

There are several issues that require further research. For example, dispatching rules that significantly improve the expected profit compared to the FIFO rule should be developed. In addition, the advanced approach, which changes dispatching rules dynamically according to the conditions of MH availability, orders waiting for cost estimation activities in the queues, and so on, should be developed. An advanced procedure to effectively determine the threshold function $MHU_{up}(EPPC_i)$ should be devised. For example, a framework for simulation-optimization [28] could be applied to the problem. In addition, a mechanism that dynamically changes rules of order selection, i.e. the threshold function, according to the cost estimation conditions, such as order arrival intervals, expected profits in each estimate class, and so on, should be developed.

Acknowledgements. This work was supported by JSPS KAKENHI Grant Number 16K01252.

References

1. Pritchard, N., Scriven, J.: EPC Contracts and Major Projects, 2nd edn. Sweet & Maxwell, London (2011)
2. Ishii, N., Takano, Y., Muraki, M.: An order acceptance strategy under limited engineering man-hours for cost estimation in Engineering-Procurement-Construction projects. Int. J. Proj. Manag. **32**(3), 519–528 (2014)
3. AACE International: Cost Estimate Classification System – As Applied in Engineering, Procurement, and Construction for the Process Industries. AACE International Recommended Practice No. 18R-97 (2011)
4. Kerzner, H.: Project Management: A Systems Approach to Planning, Scheduling, and Controlling. Wiley, Hoboken (2013)
5. Ishii, N., Takano, Y., Muraki, M.: A dynamic scheduling problem for estimating project cost. In: Proceedings of Scheduling Symposium 2015, Tokyo, pp. 119–124 (2015)
6. Ishii, N., Takano, Y., Muraki, M.: A dynamic scheduling problem in cost estimation process of EPC projects. In: Proceedings of the 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, Lisbon, Portugal, pp. 187–194 (2016)
7. Ballesteros-Pérez, P., González-Cruz, M.C., Cañavate-Grimal, A.: On competitive bidding: scoring and position probability graphs. Int. J. Proj. Manag. **31**(3), 434–448 (2013)
8. Friedman, L.: A competitive-bidding strategy. Oper. Res. **4**, 104–112 (1956)
9. Oberlender, G.D., Trost, S.M.: Predicting accuracy of early cost estimates based on estimate quality. J. Constr. Eng. Manag. **127**, 173–182 (2001)
10. Bertisen, J., Davis, G.A.: Bias and error in mine project capital cost estimation. Eng. Econ. **53**, 118–139 (2008)
11. Jørgensen, M., Halkjelsvik, T., Kitchenham, B.: How does project size affect cost estimation error? Statistical artifacts and methodological challenges. Int. J. Proj. Manag. **30**(7), 839–849 (2012)

12. Uzzafer, M.: A contingency estimation model for software projects. Int. J. Proj. Manag. **31** (7), 981–993 (2013)
13. Humphreys, K.K.: Project and Cost Engineers' Handbook. CRC Press, Boca Raton (2004)
14. Towler, G.R., Sinnott, R.: Chemical Engineering Design Principles, Practice and Economics of Plant and Process Design. Elsevier, Amsterdam (2008)
15. Rothkopf, M.H., Harstad, R.M.: Modeling competitive bidding: a critical essay. Manag. Sci. **40**, 364–384 (1994)
16. Kortanek, K.O., Sodeni, J.V., Sodaro, D.: Profit analyses and sequential bid pricing models. Manag. Sci. **20**, 396–417 (1973)
17. Ishii, N., Takano, Y., Muraki, M.: A heuristic bidding price decision algorithm based on cost estimation accuracy under limited engineering man-hours in EPC projects. In: Obaidat, M., Koziel, S., Kacprzyk, J., Leifsson, L., Ören, T. (eds.) Simulation and Modeling Methodologies, Technologies and Applications. Advances in Intelligent Systems and Computing, vol. 319, pp. 101–118 (2015)
18. Ishii, N., Takano, Y., Muraki, M.: A revised algorithm for competitive bidding price decision under limited engineering man-hours in EPC projects. Oukan (J. Transdisciplinary Fed. Sci. Technol.) **10**(1), 47–56 (2016)
19. Takano, Y., Ishii, N., Muraki, M.: A sequential competitive bidding strategy considering inaccurate cost estimates. Omega **42**(1), 132–140 (2014)
20. Takano, Y., Ishii, N., Muraki, M.: Multi-period resource allocation for estimating project costs in competitive bidding. Central Eur. J. Oper. Res. **25**(2), 303–323 (2017)
21. Shafahi, A., Haghani, A.: Modeling contractors' project selection and markup decisions influenced by eminence. Int. J. Proj. Manag. **32**(8), 1481–1493 (2014)
22. Pinedo, M.L.: Scheduling: Theory, Algorithms, and Systems, 4th edn. Springer, New York (2014)
23. Jacobs, F.R., Berry, W.L., Whybark, C.D., Vollmann, T.E.: Manufacturing Planning and Control for Supply Chain Management. McGraw-Hill, New York (2011)
24. Ishii, N.: Dynamic Scheduling Problem on Service Operations. Science Report of Research Institute for Engineering Kanagawa University, vol. 39, pp. 9–14 (2016)
25. Ishii, N., Muraki, M.: An extended dispatching rule approach in on-line scheduling framework for batch process management. Int. J. Prod. Res. **34**(2), 329–348 (1996)
26. Pritsker, A.A.B., O'Reilly, J.J.: AWESIM: the integrated simulation system. In: Proceedings of the 1998 Winter Simulation Conference, pp. 249–255 (1998)
27. Nelson, B.L., Swann, J., Goldsman, D., Song, W.: Simple procedures for selecting the best simulated system when the number of alternatives is large. Oper. Res. **49**(6), 950–963 (2001)
28. Boesel, J., Nelson, B.L., Ishii, N.: A framework for simulation-optimization software. IIE Trans. **35**(3), 221–229 (2003)

Future Prediction of Regional City Using Causal Inference Based on Time Series Data

Katsuhito Nakazawa(✉), Tetsuyoshi Shiota, and Tsutomu Tanaka

Fujitsu Laboratories Ltd., Kawasaki, Japan
{k.nakazawa,shiota,tanaka.tsuu3}@jp.fujitsu.com

Abstract. Regional cities in Japan have a lot of social issues. Various measures are being considered to solve these social issues, but it is difficult to ascertain and implement practical and effective measures. In this study, we proposed a new causal inference for selecting indicators that have causal relations with the social issues. If there was a causal relation between two sets of time series data, the slope of the approximation line of the time-shifted correlation coefficients at the base time returned a negative value. The causal inference was verified by using samples of time series data. In addition, we achieved future predictions by the vector autoregressive model using the causal indicators. The model was verified using the actual time series data of 87 regional cities. As a result, it was possible to simulate future predictions and to calculate the effects by introducing practical and effective measures to solve social issues.

Keywords: Causal inference · Future prediction · Time series data · Regional city · Social issue

1 Introduction

Regional cities in Japan have a lot of social issues such as depopulation, a decreasing birth rate and aging populations, and decline of regional industries. Globally, employment, public security, and traffic jam by urbanization are common social issues in the world. Various measures are being considered to solve these social issues, but it is difficult to ascertain and implement practical and effective measures to address them. If indicators related to the measures that have causal relations with these issues can be determined through data-based analyses, more practical and effective measures can be employed. Though several causal inferences using statistical analysis have been proposed so far, they prove the causal relations of already-known incidents based on their hypothesis [1–3]. We should find indicators that have causal relations with social issues from many and unspecified data.

In this study, our objectives are to propose a new causal inference for selecting indicators that have causal relations to solve social issues, and to achieve future predictions for regional cities using the causal indicators and to quantify the effects of introduced measures. As a result, it will be possible for regional governments to plan the practical and effective measures that originate in the causal indicators obtained from this

M.S. Obaidat et al. (eds.), *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing 676,
https://doi.org/10.1007/978-3-319-69832-8_16

causal inference. In addition, they will be able to make predictions regarding their regional cities in the future according to a model using the indicators that have causal relations with various social issues.

2 Causal Inference Using Time Series Data

We considered that time series data were useful to determine causal relations because the causal indicators and the effect indicators were distinguished easily by shifting the time of two time series data sets. The Granger causality concept is already well-known for determining causality using time series data [4]. However, it is difficult to explain the causal relation between two time series data sets for a short term. Though the Convergent Cross Mapping is a good method to find causality in nonlinear time series data, it is unsuitable for the determining of social time series data that are mostly linear relations [5].

"e-Stat", a portal site in Japan, releases various time series data for 1,742 Japanese regional cities [6]. The term of the time series data investigated every year is mainly from 2000 to 2013, and we need a new causal inference using the time series data for the short time period to find various indicators that have causal relations with social issues.

In this work, we propose a causal inference using time series data to plan practical and efficient measures for regional cities and to carry out future predictions by models using the indicators that have causal relations with social issues. The following hypothesis of the causal inference was conceived in this study.

2.1 Hypothesis of Causal Inference Using Time Series Data

In this study, we proposed a causal inference to find out the causal relations from variation of correlation coefficient between two indicators by shifting the time of two sets of time series data. It is a method to determine the causal relations based on the Pearson product-moment correlation coefficient [7–9].

According to the Pearson product-moment correlation coefficient, the correlation coefficient R of Indicator X and Indicator Y is calculated from the following Eq. (1):

$$R = \frac{\sum_{i=1}^{n} \left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)}{\sqrt{\sum_{i=1}^{n} \left(x_i - \bar{x}\right)^2} \sqrt{\sum_{i=1}^{n} \left(y_i - \bar{y}\right)^2}} \tag{1}$$

where x indicates the time series data of Indicator X and y indicates the time series data of Indicator Y.

Expressing this with the average Eq. (2) of the time series data of indicator X and the average Eq. (3) of the time series data of indicator Y gives us following Eq. (4).

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{2}$$

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{3}$$

$$R = \frac{\sum_{i=1}^{n} x_i y_i - \frac{1}{n}\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2}\sqrt{\sum_{i=1}^{n} y_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} y_i\right)^2}} \tag{4}$$

Then, expressing Eq. (4) with Eqs. (5), (6), (7), (8) and (9) respectively, gives us Eq. (10).

$$S_X = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{5}$$

$$S_Y = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{6}$$

$$S_{XX} = \frac{1}{n}\sum_{i=1}^{n} x_i x_i \tag{7}$$

$$S_{YY} = \frac{1}{n}\sum_{i=1}^{n} y_i y_i \tag{8}$$

$$S_{XY} = \frac{1}{n}\sum_{i=1}^{n} x_i y_i \tag{9}$$

$$R = \frac{S_{XY} - \frac{1}{n}S_X S_Y}{\sqrt{S_{XX} - \frac{1}{n}S_X^2}\sqrt{S_{YY} - \frac{1}{n}S_Y^2}} \tag{10}$$

Simple time series data of Indicator X shown in Table 1 were prepared to prove the hypothesis. These time series data City A, B and C change in three patterns from Time $T1-2$ to Time $T1+1$.

Table 1. Time series data of Indicator X.

t	$T1-2$	$T1-1$	$T1$	$T1+1$
City A	$x-1$	x	$x+1$	$x+2$
City B	x	x	x	x
City C	$x+3$	$x+2$	$x+1$	x

If Indicator Y has a causal relation with Indicator X completely ($R = 1.0$), the time series data of Indicator Y are shown as Table 2 according to the regression line:

$Y(t) = aX(t-1) + b$ $(a > 0)$. We assumed simply that Indicator X influences Indicator Y at the next unit time.

Table 2. Time series data of Indicator Y.

t	T1 − 1	T1	T1 + 1
City A	a(x − 1) + b	ax + b	a(x + 1) + b
City B	ax + b	ax + b	ax + b
City C	a(x + 3) + b	a(x + 2) + b	a(x + 1) + b

From Tables 1 and 2, when Indicator Y is fixed at Time T1 and Indicator X is shifted at each Time T1 − 1, T1 and T1 + 1, Eqs. (5), (6), (7), (8) and (9) are shown in Table 3.

Table 3. Expressions of Indicator X and Indicator Y when Indicator Y is at Time T1 and Indicator X is at each Time T1 − 1, T1 and T1 + 1.

t for X	T1 − 1	T1	T1 + 1
S_X	(3x + 2)/3	(3x + 2)/3	(3x + 2)/3
S_Y	(3y + 2a)/3	(3y + 2a)/3	(3y + 2a)/3
S_{XX}	$(3x^2 + 4x + 4)/3$	$(3x^2 + 4x + 2)/3$	$(3x^2 + 4x + 4)/3$
S_{YY}	$(3y^2 + 4ay + 4a^2)/3$	$(3y^2 + 4ay + 4a^2)/3$	$(3y^2 + 4ay + 4a^2)/3$
S_{XY}	(3xy + 2y + 2ax + 4a)/3	(3xy + 2y + 2ax +2a)/3	(3xy + 2y + 2ax)/3

Equation (10) provides the correlation coefficient by substituting the equations of Table 3. Figure 1 shows the correlation coefficient of Indicator X to Indicator Y at Time T1. This is the representative result as it is not dependent on the values of a and b in $Y(t) = aX(t-1) + b$. From this result, we built the following hypothesis of the causal inference: if Indicator Y has a causal relation with Indicator X and Indicator X is the causal indicator of Indicator Y, $R_{T1-1} > R_{T1} > R_{T1+1}$ is completed.

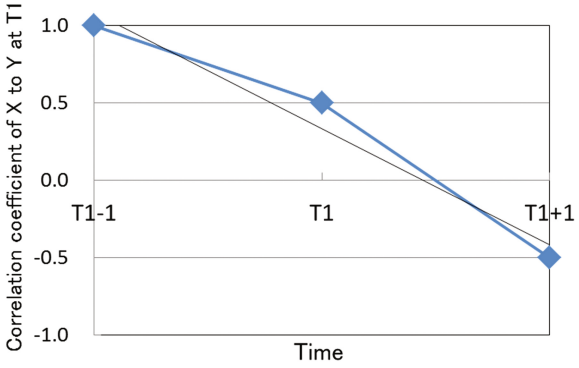


Fig. 1. Correlation coefficient of Indicator X to Indicator Y at Time T1 by the Pearson product-moment correlation coefficient.

In other words, the correlation coefficients of Indicator X to Indicator Y at a base time: T become lower as shown in Fig. 2(a), and the slope of the approximation line has a negative value. Similarly, the correlation coefficients of Indicator Y to Indicator X at a base time: T rise, as shown in Fig. 2(b) if Indicator X is the causal indicator of Indicator Y, and the slope of the approximation line has a positive value.

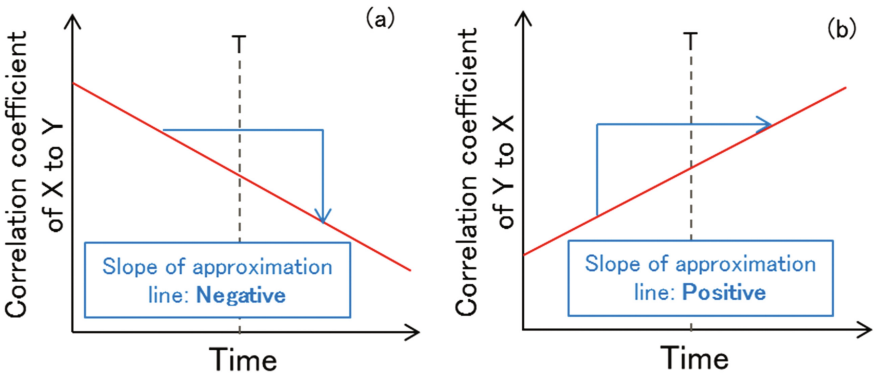


Fig. 2. Correlation coefficient of Indicator X to Indicator Y at base time T (a) and correlation coefficient of Indicator Y to Indicator X at base time T (b) when Indicator X is a causal indicator of Indicator Y.

2.2 Verification of Causal Inference Using Samples of Time Series Data

The causal inference was applied to samples of time series data with an already-known causal relation to confirm the above-mentioned hypothesis. The samples of time series data of the causal indicator in this verification are shown in Fig. 3.

We assumed and prepared the samples of time series data of the causal indicator for 10 cities from City D to City M between 1990 and 2010. The data of each city increased 10% per year for 10 years and decreased 10% per year for 10 years, and the changing years and the initial values of the 10 cities were different respectively.

Next, we assumed that the samples of time series data of the causal indicator affected samples of time series data of the effect indicator in the next year by 30%. And the influence of time series data of the causal indicator gradually decreased by 3% every year, which lasted for 10 years. The samples of time series data of the effect indicator for 10 cities from City D to City M between 2000 and 2010 in this verification are shown in Fig. 4.

We calculated the correlation coefficients using the sample data of the effect indicator based on 2005 and the samples of time series data of the causal indicator between 2000 and 2010. From the result of Fig. 5, if there is a causal relation between two sets of time series data, the slope of the approximation line of the correlation coefficients at the base time returns a negative value, and our hypothesis could be proved by using samples of time series data.

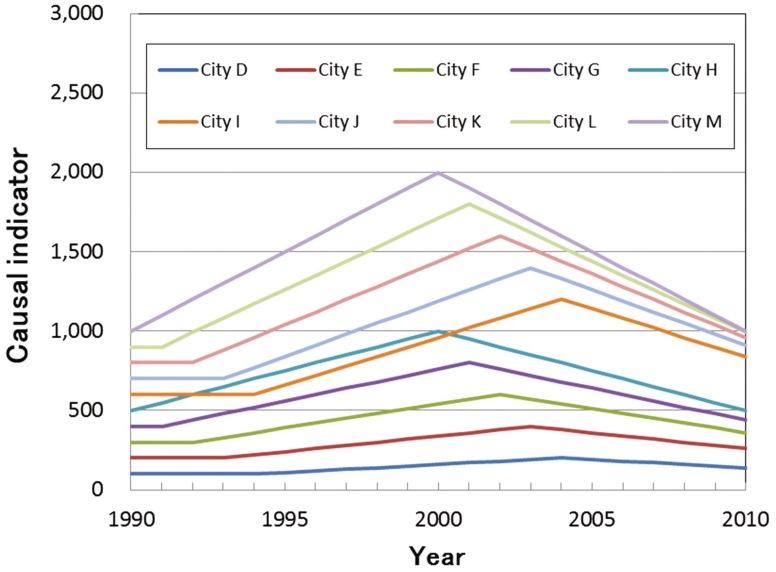


Fig. 3. Samples of time series data of causal indicator for 10 cities between 1990 and 2010.

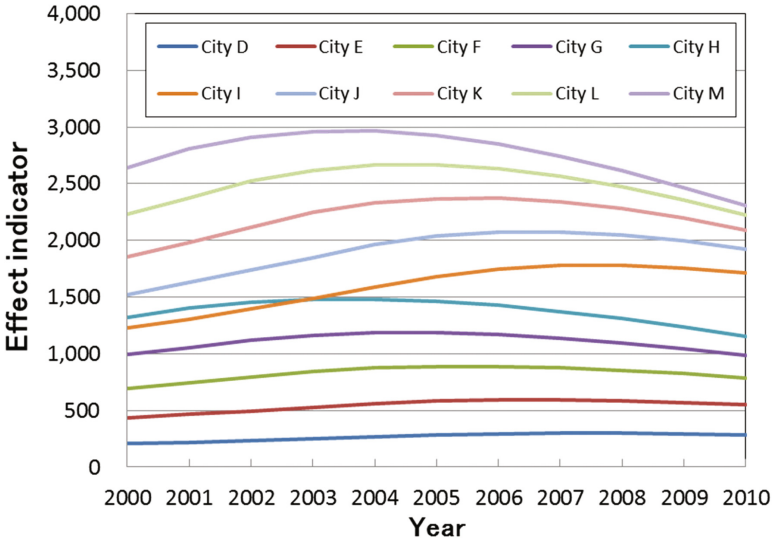


Fig. 4. Samples of time series data of effect indicator for 10 cities between 2000 and 2010.

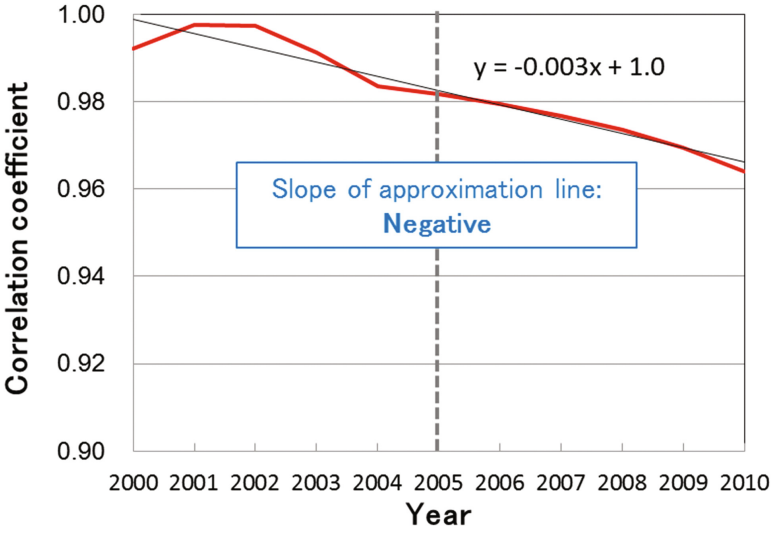


Fig. 5. Correlation coefficients using sample data of effect indicator based on 2005 and samples of time series data of causal indicator between 2000 and 2010.

3 Simulation Model for Future Prediction

We considered that future predictions of regional cities could be conducted by selecting an appropriate model using the causal indicators. As mentioned below, several simulation models for the future predictions were verified, and the most suitable model was selected.

3.1 Selecting Simulation Model Through Verification

The model selection was considered using the following simulation models, verification method, and time series data.

3.1.1 Simulation Models

As models in which plural causal indicators as explanatory variables were available, the following 3 types of regression models:

1. Multivariate regression model (MR model)
2. Stepwise regression model (SW model)
3. Vector autoregressive model (VAR model)

were verified in this study [10].

3.1.2 Verification Method

Population issue is a common significant target for a lot of regional cities in Japan. We predicted the total population from 2000 to 2013 using time series data from 1985 to 1999 by 3 types of the simulation models, compared with actual data from 2000 to 2013. In this verification, an actual city (City N) of 1.2 million population scale was targeted.

3.1.3 Time Series Data

First, we constructed a network of causal indicators based on the above-mentioned causal inference. 238 kinds of time series data between 2000 and 2013 for 1,742 regional cities in Japan were included in this network, and the causal relations were mutually calculated using the causal inference. Causal indicators can be easily found and selected using this network.
The following 6 kinds of time series data:

- Live births (person)
- In-migrants from other prefectures (person)
- Kindergarten pupils (person)
- Marriages (couple)
- Taxable income (thousand yen)
- Tax debtors per income levy (person)

were selected as the causal indicators of the total population from the network of 238 kinds of time series data. We also conducted the verification using the time series data that directly influenced the total population such as the deaths and the out-migrants to other prefectures in addition to the live births and the in-migrants. The time series data of City N for this verification are shown in Table 4.

Table 4. Time series data of causal and direct indicators from 1985 to 1999 of City N for verification of each simulation model.

Year	Total population (person)	Live births (person)	In-migrants (person)	Marriages (couple)	Taxable income (million yen)	Tax debtors (person)	Kindergarten pupils (person)	Deaths (person)	Out-migrants (person)
1985	1,088,624	14,003	83,718	8,697	1,317,664	443,164	21,452	4,477	78,451
1986	1,106,148	13,773	87,562	8,522	1,410,422	455,918	21,317	4,523	78,085
1987	1,126,485	13,999	90,742	8,885	1,521,780	471,283	21,790	4,753	80,193
1988	1,142,953	13,920	88,421	9,166	1,696,876	487,709	22,004	5,060	81,131
1989	1,157,005	13,090	91,848	9,484	1,793,159	496,645	21,918	5,038	85,576
1990	1,173,603	13,279	93,797	9,696	2,030,951	505,254	21,515	5,346	86,633
1991	1,187,034	13,494	91,537	10,049	2,243,247	528,811	21,582	5,487	87,751
1992	1,195,464	13,356	91,587	10,226	2,407,043	545,002	21,254	5,736	91,665
1993	1,199,707	12,855	90,167	10,718	2,341,293	557,276	20,895	6,032	93,102
1994	1,202,069	13,476	89,639	10,857	2,378,228	561,607	19,952	6,153	94,026
1995	1,202,820	13,146	87,846	10,897	2,398,720	561,574	19,476	6,399	91,268
1996	1,209,212	13,309	88,284	11,147	2,381,925	564,303	19,673	6,265	88,317
1997	1,217,359	13,423	87,209	10,465	2,443,415	567,349	19,799	6,461	85,304
1998	1,229,789	13,756	88,702	10,759	2,459,855	572,562	20,565	6,783	83,223
1999	1,240,172	13,590	87,196	10,211	2,417,085	572,331	21,071	7,186	83,975

3.2 Verification of Future Prediction Using Simulation Models

The verification result using 3 types of simulation models and using the 8 kinds of time series data is shown in Fig. 6. In this verification, we tried out 5 types of simulation:

- VAR model using 6 kinds of causal indicators such as live births, in-migrants, kindergarten pupils, marriages, taxable income, and tax debtors as explanatory variables
- VAR model using 4 kinds of direct indicators such as deaths, out-migrants, live births, and in-migrants as explanatory variables
- MR model using 6 kinds of causal indicators such as live births, in-migrants, kindergarten pupils, marriages, taxable income, and tax debtors as explanatory variables
- MR model using 4 kinds of direct indicators such as deaths, out-migrants, live births, and in-migrants as explanatory variables
- SW model using in-migrants, deaths, tax debtors, and kindergarten pupils that were chosen as effective explanatory variables

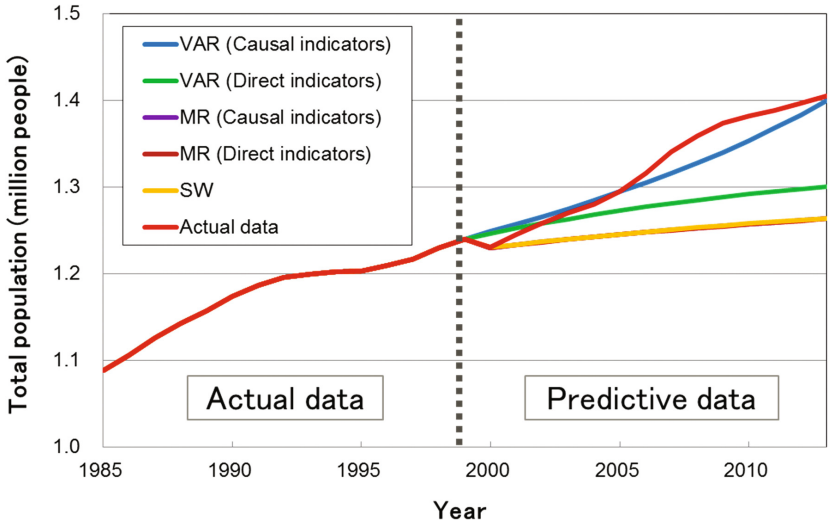


Fig. 6. Verification of total population by 5 types of simulation models using different kinds of time series data. The results of MR models are almost the same level as the result of SW model.

As a result, we observed that the simulations using the VAR models showed good coincidence with the actual data, especially in terms of using the causal indicators. The average rate of relative error for the actual data with the VAR model using the causal indicators was the lowest by 1.1%/year. The next lowest simulation was the VAR model using the direct indicators, and the average relative error was 3.7%/year. The average relative errors in the other simulation models were 5.6%/year at the same level.

We confirmed the relationship between the number of actual time series data and the average rate of relative error of the simulation by the VAR model using the causal indicators. Figure 7 shows the average rate of relative error for each year prediction of City N by VAR model using each number of actual time series data. From this result,

we confirmed the tendency that the average rate of relative error rose as the actual time series data were short. In this case, it is necessary to collect the number of actual time series data to 10 or more in order to adjust the average rate of relative error to 10% or less.

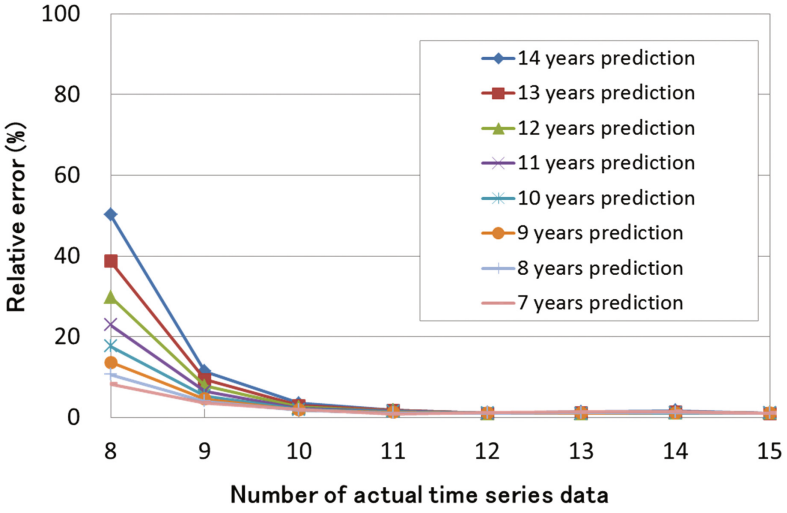


Fig. 7. Relationship between the number of actual time series data for each year prediction and relative error in case of City N.

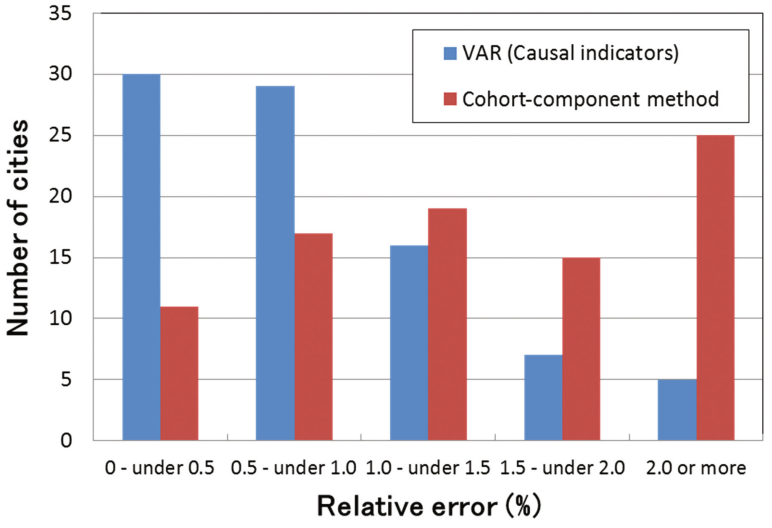


Fig. 8. The number of cities in each average rate of relative error of total population by VAR model and cohort-component method.

Next, We calculated the average rate of relative error of simulations by the VAR model using the causal indicators, compared with the cohort-component method. The future population investigated by the cohort-component method is used as a bench mark in Japan [11]. In this comparison, 87 cities that were 5% cities divided into 10 categories on Japanese population scale were selected. The total population from 2007 to 2013 was predicted by the VAR model using time series data from 2000 to 2006 of the 6 kinds of causal indicators. Figure 8 shows the number of cities in each average rate of relative error of the population prediction by both methods. It was confirmed that the total population by the VAR model using the causal indicators was predictable in a smaller relative error.

We concluded that it was possible to predict regional cities in the future by the vector autoregressive model using the causal indicators that have causal relations with social issues.

4 Future Prediction of Regional City Using Causal Indicators

A lot of regional cities in Japan are experiencing depopulation issues as described above, We predicted the future population of a regional city by the VAR model using the causal indicators.

4.1 Future Prediction for Regional City in the Future

We selected the causal indicators of the live births, the in-migrants, the kindergarten pupils, the marriages, the taxable income, and the tax debtors per income levy that have causal relations with the total population. Figure 9 shows the causal relation diagram of

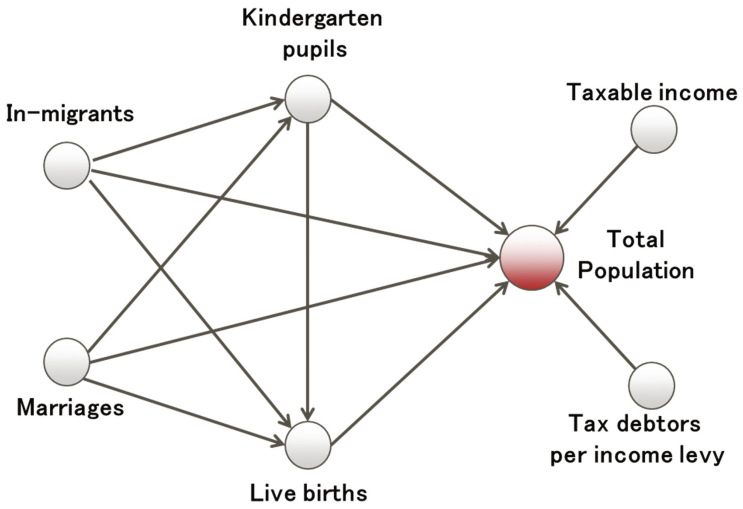


Fig. 9. Causal relation diagram of total population and causal indicators selected from network of 238 kinds of time series data.

total population and the causal indicators. These causal indicators that have correlation coefficients of 0.9 or more based on the total population data in 2006 were selected from the network of 238 kinds of time series data. We predicted the total population from 2014 to 2050 by the VAR model using actual time series data of the causal indicators from 2000 to 2013. In this future prediction, City O with a population of 275,000 was targeted.

The future prediction of total population in City O was shown in Fig. 10. As a result, we predicted that the total population of City O would decrease from 276,000 people to 258,000 people between 2014 and 2050 (the population decrease rate being 6.3%). It was suggested that the decrease of the total population was one of the social issues for City O as well as a lot of regional cities in Japan.

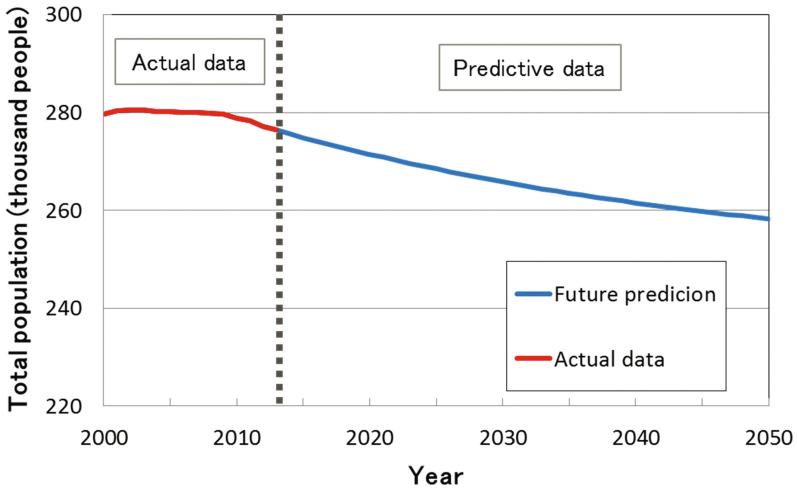


Fig. 10. Future prediction of total population in City O from 2014 to 2050 by VAR model using causal indicators.

4.2 Future Prediction of Total Population by Introducing Measure

As shown in Fig. 9, in-migrants were one of important causal indicators of total population from the viewpoint of population growth in the future. Then, we verified future predictions by actual case studies introducing a measure of in-migrants increase. 4 regional cities from City P to City S that introduced a measure to increase in-migrants were selected in this verification. In these cities, the measure that the residential area was increased by the land development was introduced in 2006 or 2007, and in-migrants consequently increased by 68%, 25%, 66% and 38% in 2013, respectively. The total populations from 2007 or 2008 to 2013 were verified by the VAR model using actual time series data of in-migrants from 2007 or 2008 to 2013, compared with actual data. Figure 11 shows the verification results of 4 regional cities introducing the measure to increase in-migrants. Each average rate of relative error for the actual data was 1.2%/year in City P, 3.0%/year in City Q, 3.6%/year in City R and 2.6%/year in City S, and

we confirmed the total population introducing the measure by the VAR model was predictable in a smaller relative error.

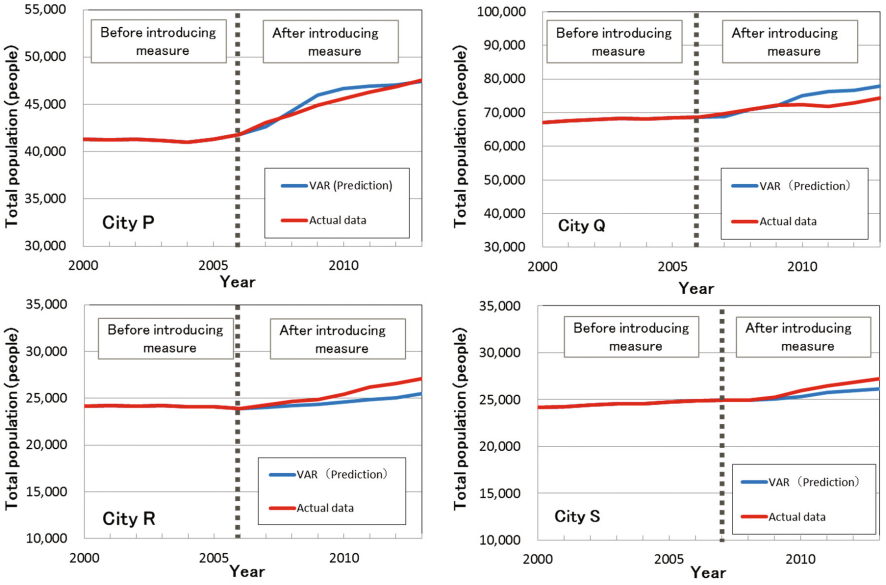


Fig. 11. Verification of simulations for total population in 4 regional cities by introducing measure for in-migrants, compared with actual data.

Next, we assumed a measure for increasing in-migrants in City O, and set 5 types of scenario that increase in-migrants gradually from 2017 to 2025, and increase from 2026 to 2050 by 10% (Scenario A1), 20% (Scenario A2), 30% (Scenario A3), 40% (Scenario A4) and 50% (Scenario A5), compared with the BAU scenario. Figure 12 shows the simulation result of the total population in each scenario. From this result, we confirmed that total population in City O increased by increasing in-migrants between 2017 and 2050. In the case of Scenario A4 (increasing by 40% from 2026 to 2050), the current total population in City O (275,000 people) could be maintained in 2050.

Similarly, we considered that the kindergarten pupils and the live births were significant indicators to increase the total population for City O. Then, the simulations of the total population were conducted by introducing measures to improve the kindergarten pupils and the live births.

As well as the case of the in-migrants, we assumed measures for increasing the kindergarten pupils and the live births, and set 5 types of scenarios respectively that increase the kindergarten pupils and the live births gradually from 2017 to 2025, and increase from 2026 to 2050 by 10% (Scenario B1 and C1), 20% (Scenario B2 and C2), 30% (Scenario B3 and C3), 40% (Scenario B4 and C4) and 50% (Scenario B5 and C5), compared with the BAU scenario.

Figure 13 shows the simulation results of the total population by introducing measure for the kindergarten pupils in each scenario. From this result, it was confirmed that the

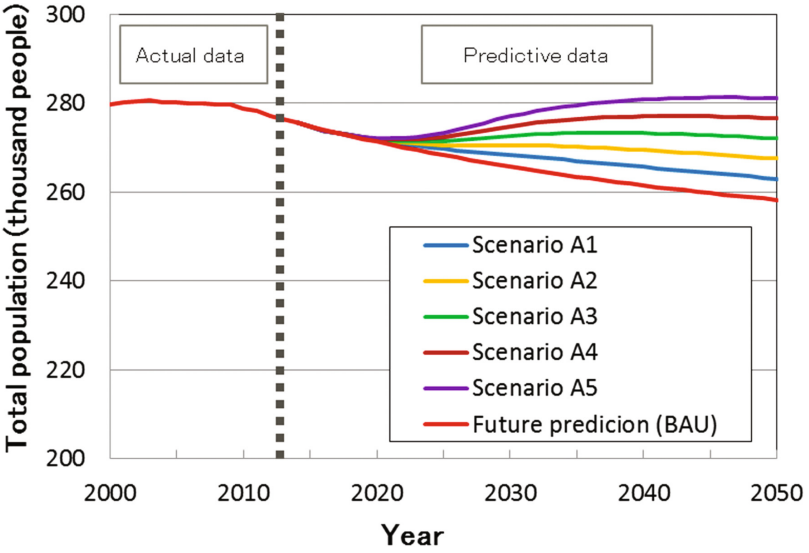


Fig. 12. Simulation results of total population in 5 scenarios by introducing measure for in-migrants in City O.

total population increased by increasing kindergarten pupils, and the current total population in City O could be maintained in 2050 in the case of Scenario B4.

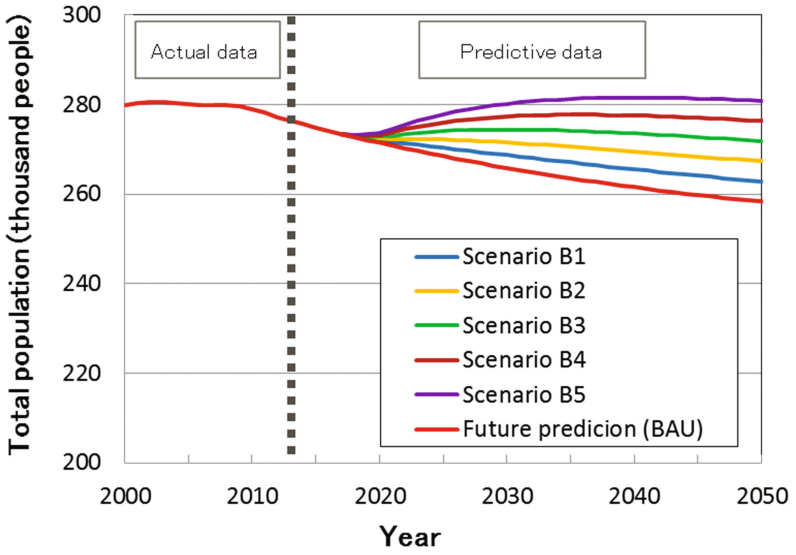


Fig. 13. Simulation results of total population in 5 scenarios by introducing measure for kindergarten pupils in City O.

Figure 14 shows the simulation results of the total population by introducing measure for the live births in each scenario. From this result, the total population increased by increasing the live births, and the current total population in City O could be maintained in 2050 in the case of Scenario C5.

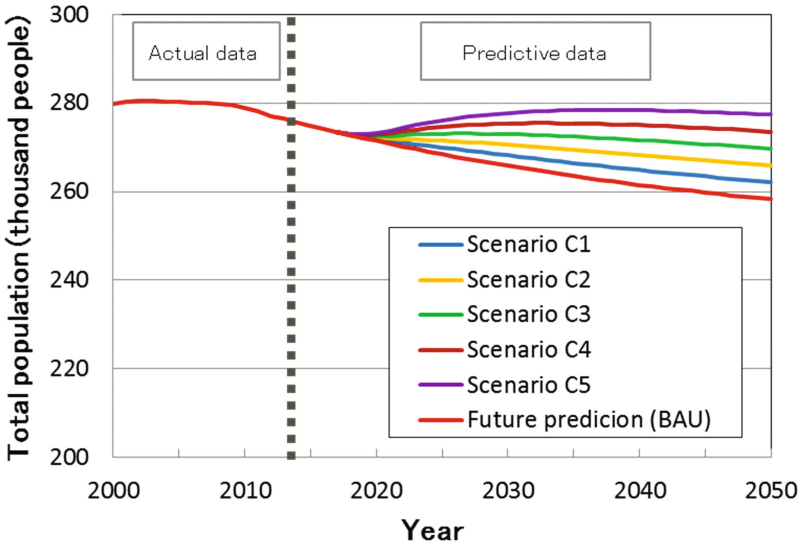


Fig. 14. Simulation results of total population in 5 scenarios by introducing measure for live births in City O.

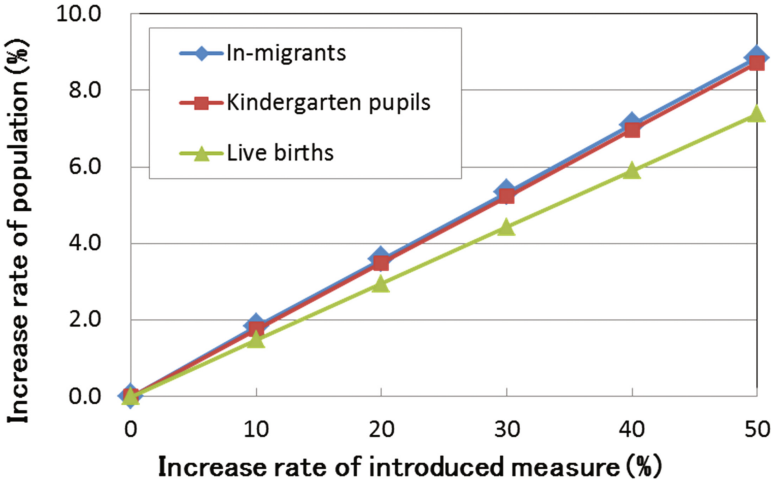


Fig. 15. Relationship between increase rate of total population and increase rate of introduced measure in City O.

Figure 15 shows the relationship between the increase rate of the total population and the increase rate of measures introduced for the in-migrants, the kindergarten pupils, and the live births. We obtained the result that it is necessary to increase the in-migrants by 5.5%, the kindergarten pupils by 5.7% and the live births by 6.8% for increasing the total population by 1.0% for the BAU scenario.

It was suggested that the measure for the in-migrants could be more effective in increasing the total population of City O if the difficulties such as cost-effectiveness, resident acceptability and feasibility in introducing these measures were at the same level.

5 Conclusions

We proposed a causal inference for selecting indicators that have causal relations with social issues. If there was a causal relation between two sets of time series data, the slope of the approximation line of the time-shifted correlation coefficients at the base time returned a negative value. The causal inference was verified by using samples of time series data. In addition, we also achieved future predictions by the vector autoregressive model using the causal indicators. The model was verified using the actual time series data of 87 regional cities in Japan. As a result, it was possible to simulate future predictions and to calculate the effects by introducing practical and effective measures for the in-migrants, the kindergarten pupils, and the live births that originated from the social issue with deceasing total population.

As mentioned above, it was easily possible to determine the causal indicators and to quantify the effect of introducing the measures by the VAR model using the causal indicators in this study. For future work, we will be able to apply this causal inference to a number of social issues by including more indicators related to economic and environmental time series data in the network, and expand it to various fields. Moreover, we need to simulate and verify the effects of introducing measures in consideration of the characteristics of regional cities.

In terms of establishing a sustainable society, we expect that regional governments select appropriate measures based on causal inferences through data-based analyses, and decide on optimal measures after executing future predictions when these measures are introduced.

References

1. Rubin, D.: Estimating causal effects of treatments in randomized and nonrandomized studies. J. Educ. Psychol. **66**(5), 688–701 (1974)
2. Pearl, J.: Bayesian networks: a model of self-activated memory for evidential reasoning. In: Proceedings, Cognitive Science Society, pp. 329–334 (1985)
3. Shimizu, S., Hyvärinen, A., Kano, Y., Hoyer, P.O.: Discovery of non-Gaussian linear causal models using ICA. In: Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence, UAI 2005, pp. 526–533 (2005)
4. Granger, C.W.J.: Investigating causal relations by econometric models and cross-spectral methods. Econometrica **37**(3), 424–438 (1969)

5. Sugihara, G., May, R., Ye, H., Hsieh, C., Deyle, E., Fogarty, M., Munch, S.: Detecting causality in complex ecosystems. Science **338**, 496–500 (2012)
6. Ministry of Internal Affairs and Communications: e-Stat: portal site of official statistics of Japan (2016). http://www.e-sat.go.jp/SG1/estat/eStatTopPortalE.do
7. Rodgers, J.L., Nicewander, W.A.: Thirteen ways to look at the correlation coefficient. Am. Stat. **42**(1), 59–66 (1988)
8. Rderrick, T., Tbates, B., Sdufek, J.: Evaluation of time-series data sets using the Pearson product-moment correlation coefficient. Med. Sci. Sports Exerc. **26**(7), 919–928 (1994)
9. Mrobinson, P.: Correlation testing in time series, spatial and cross-sectional data. J. Econometrics **147**(1), 5–16 (2008)
10. Sims, C.A.: Macroeconomics and reality. Econometrica **48**, 1–48 (1980)
11. National Institute of Population and Social Security Research: Regional population projections for Japan: 2010–2040, Population Research Series, p. 330 (2013)

Cooperative Radio Resources Allocation and Congestion Prevention Scheme for LTE-A

Mzoughi Houda[1(✉)], Faouzi Zarai[1], Mohammad S. Obaidat[2], Balqies Sadoun[3], and Lotfi Kamoun[1]

[1] LETI Laboratory, University of Sfax, Sfax, Tunisia
Houda_mzoughi_enis@yahoo.fr,
Faouzi.zarai@isecs.rnu.tn
[2] King Abdullah II School of Information Technology, The University of Jordan, Amman, Jordan
msobaidat@gmail.com
[3] College of Engineering, Al-Balqa' Applied University, Al-Salt, Jordan
faouzifbz@gmail.com

Abstract. The objective of this paper is to define new radio resources allocation scheme in heterogeneous environment that includes congestion control in LTE-A cells. The originality of the proposed approach is that it manages both mobile users and physical resources block. In addition, it is based on the deployment of the Media Independent Handover protocol in order to facilitate the communication between different network entities when performing inter cell cooperation as well as when undertaking handover process. The proposed scheme is evaluated with simulation analysis and results show throughput improvement.

Keywords: LTE-A · MIMO/OFDMA · Radio resources allocation · Handover · MIH · Modeling and simulation · Performance evaluation

1 Introduction

New generations wireless mobile communication systems, provide to end users several multimedia services. Many of these services are expensive in terms of resources. In order to deal with this explosive resources demands, it is essential to have a large network capacity. In order to fulfill such requirement, some technologies are chosen to be deployed for the new generation wireless network, like the LTE-A systems, such as MIMO, OFDMA, Beamforming, small cells enhancements, macro cells enhancements, HetNets, etc. Coming along with their benefits, these new technologies introduce new challenges in radio resource management (RRM) feature, which is vital to ensure efficient exploitation of the available radio resources including interference management and resource allocation. In this work, we deal with radio resource allocation in downlink in the LTE-A system while ensuring cooperation aspect inside system [1].

Regarding the literature, we find out several interesting works investigating the problem of radio resource allocation in OFDMA networks, some of which are mentioned below. Some works investigated resource allocation based on optimization

M.S. Obaidat et al. (eds.), *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing 676,
https://doi.org/10.1007/978-3-319-69832-8_17

theory approach [2–5], and others formulated the problem based on a game theory approach [6, 7]. In [2] a radio resources allocation scheme for MIMO/OFDMA systems with the employment of Beamforming technique is suggested. In this work, end users are classified into two groups: interior users and exterior users in the cell. Interior users are those for which the interference term satisfies that the sum of received signals in all beams is considerably smaller than the Gaussian noise; all others are considered as exterior users. Then, the total number of PRB is divided into Q groups (GPRB) with an equal size, to be next sorted in decreasing order of power. The set of GPRBs that will be allocated to exterior users, which is calculated according to their number in the cell, are those with the minimum power for all eNodeBs. The rest of Q GPRBs will be automatically allocated to interior users. In [3], authors considered imperfect channel state information (CSI). This means that the CSI is estimated at the receiver, then it is fed back to the source. The proposed scheme takes into account the bit error rate, the rate requirement and delay requirement as QoS constraints. They proposed heuristic approach with three steps. In the first step, the resource allocation unit decides the number of subcarriers needed by each user according to its QoS requirements. Secondly, subcarriers are assigned to users according to the corresponding power allocation based on the outcomes of the first step and power constraint. In the third step, they suggested the reallocation of some subcarriers according to user's satisfaction.

2 The Proposed Cooperative Radio Resources Allocation

In this section, we present a dynamic and cooperative solution to the problem in order to satisfy the LTE-A network features. We considered both network and users' constraints. We classify users into edge and central users. In both cases, users will be collected to MU-teams according to their localization, their velocity and their QoS requirements. In the other side, PRBs are assigned by MU-team, we estimated the required number of PRBs by each MU-team. Then the total available band is divided into sub-bands according to the estimated value of required PRBs, one of them will be assigned to the corresponding MU-team. The sub-band selection is based on the resolution of capacity maximization problem. We also studied the case when the MU-team requirement risks overloading the cell, in such case the cell will collaborate/coordinate with neighbors one to serve the group of users.

When working in this contribution, we look to find a solution of the problem, while maximizing system capacity and maintaining performance. As mentioned above, we focus on LTE-A RAT, but it could be extended to other RTAs. Our approach includes four steps. In the first step, we focus on active users classified into central and edge users. For both cases, we divide users into groups to form MU-teams. Such idea can replace the deployment of small cells in order to cancel inter small cell interference. In addition, it aims to resolve problems with edge user like interference and Ping-Pong effect. In the second step, we select radio resources that will be allocated to each MU-team, based on Channel Quality Indicator (CQI). Next, we track the allocation to check whether the MU-teams' requirements are satisfied or not, and if there is an over-served MU-team. Both cases need a reallocation to improve proposed scheme efficiency. In the last step, we aim to prevent a congestion state in the cell. So,

we propose two different means. First approach is based on have collaboration between overloaded cell with those that are under-loaded to serve some active users. The second approach is based on triggering handover session for some users. We propose also a new target selection scheme for handover session that considers several criteria, including load factor and candidate capacity.

2.1 System Model

In this contribution, we investigate radio resource allocation for multi-cell downlink OFDMA communication scenario in LTE-A systems. All eNodeBs are connected and perform joint transmission to all users; such a case illustrates the global joint transmission [8–10] (Fig. 1).

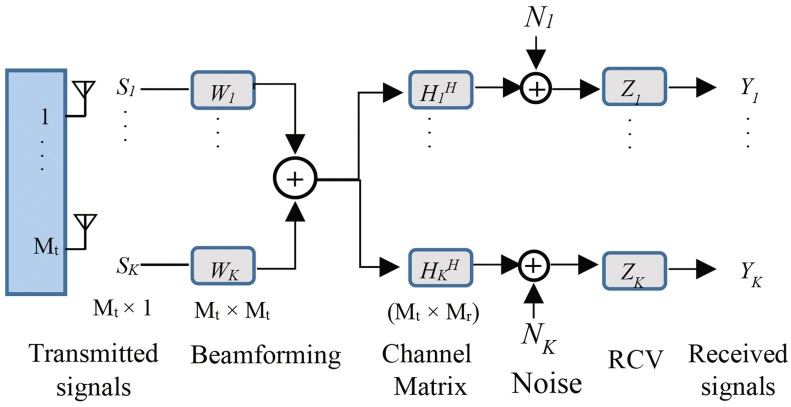


Fig. 1. Illustration of MIMO downlink communication.

With the OFDM scheme, the total band is equally divided into P physical resource blocks (PRB) and each user can allocate an integer number of PRBs according to its requirements. Consider MIMO as key technology of LTE-A, where each eNodeB is equipped with Me transmit antennas and users' devices are equipped with M_r receive antennas. Each eNodeB transmits a single data stream to each user with zero forcing beamforming [11, 12]. We mean by M_t the total number of transmit antennas equal to $\sum_{i=1}^{E} Me_i$, where E denotes the total number of eNodeB.

The received signal Y_k^p at user k in the set of K user scheduled in PRB p is modeled as shown below:

$$Y_k^p = Z_k\left(\sum_{i=1}^{E} H_{k,i}^p W_{k,i}^p S_{k,i}^p + \sum_{j=1, j\neq k}^{K}\sum_{i=1}^{E} H_{k,i}^p W_{j,i}^p S_{j,i}^p + N_k^p\right) \tag{1}$$

Where, $H_{k,i}^p \in C^{M_r \times M_e}$ is the channel matrix between user k and eNodeB i at PRB p. $S_{j,i}^p \in C^{M_e \times l}$ is the data vector transmitted by the eNodeB i for user j employing the beamforming vector $W_{j,i}^p \in C^{M_e \times l}$ with $E\left[S_{j,i}^p \left(S_{j,i}^p\right)^H\right] = 1$. N_k^p is the additive white Gaussian noise with zero mean and covariance matrix $\sigma^2 I_{\mathrm{Mr}}$. Furthermore Z_k denotes the combining receive vector employed at user k. Therefore, the SINR for user k connected to eNodeB e at PRB p is modeled as below:

$$SINR_{k,e}^p = \frac{\left|Z_k H_{k,e}^p W_{k,e}\right|^2}{\sum\limits_{\substack{i=1 \\ i\neq e}}^{E}\left|Z_k H_{k,i}^p W_{k,i}^p\right|^2 + \sum\limits_{\substack{j=1 \\ j\neq k}}^{K}\sum\limits_{i=1}^{E}\left|Z_k H_{k,i}^p W_{j,i}^p\right|^2 + \left|Z_k \sigma_k\right|^2} \tag{2}$$

Moreover, the interference expression in (2) includes the interference introduced from other eNodeB (ICI) as well as inter-user interference (co-user channel). However, with Joint Transmission, inter-eNodeBs interference can be neglected; thanks to the coordination between all eNodeBs when serving all covered users.

In practice, the uncertainty of CSI makes the cancellation of inter-users' interference an impossible task [1]. The zero-forcing beamforming strategy can give inter-users interference relaxation by being limited to some threshold value $\gamma > 0$ instead of being cancelled [13], which means that:

$$\sum_{\substack{j=1 \\ j\neq k}}^{K}\sum_{i=1}^{E}\left|Z_k H_{k,i}^p W_{j,i}^p\right|^2 \leq \gamma \tag{3}$$

According to the description given above, the SINR at user k can be formulated as:

$$SINR_{k,e}^p = \frac{\left|Z_k H_{k,e}^p W_{k,e}\right|^2}{\sum\limits_{\substack{j=1 \\ j\neq k}}^{K}\sum\limits_{i=1}^{E}\left|Z_k H_{k,i}^p W_{j,i}^p\right|^2 + \left|Z_k \sigma_k\right|^2} \tag{4}$$

The achieved data rate by user k in its served cell e for one PRB p:

$$R_{k,e} = \frac{1}{N_p} \sum_{p=1}^{N_p} \log_2(1 + SINR_{k,e}^{p}) \tag{5}$$

Where N_p denotes the number of allocated PRB to user k by eNodeB e, and the total system capacity is modeled as:

$$C_T = \sum_{e=1}^{E} \sum_{k=1}^{K} R_{k,e} \tag{6}$$

2.2 Inter-mobile Users' Cooperation

In the macro cell, users are classified into edge and central ones. Then, they are collected into groups to form MU- teams according to three metrics: localization, user's application and user's velocity. The role of team leader can be assigned to one UE at the same time according to an agreement with the operator. However, this role is not limited to a single user; another one can fill it when principal leader is down so as to maintain the MU-team. The number of mobile users by each MU-team may be fixed taking into consideration the capacities of the leader device (Fig. 2).

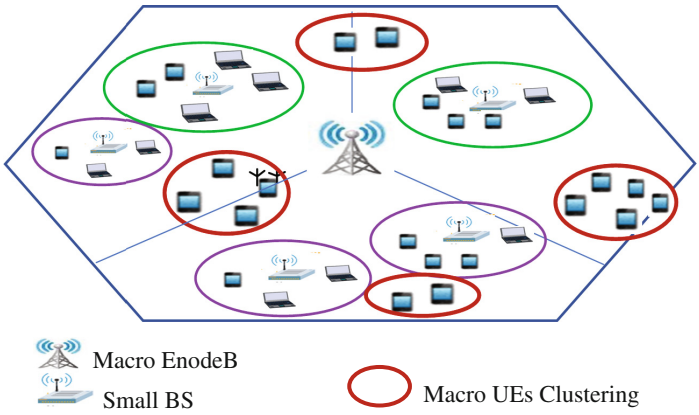


Fig. 2. Small cells with macro users clustering deployment.

The proposed algorithm for MU- teams' management is presented below.

Algorithm 1. M-UE teams Conception.

```
Initialization
L: list of active users in the sector/cell
B: number of MU-teams, B=0
V_k: velocity of user k
R_av: available data rate in the MU-team
RR_k: required rate by user k
# Create MU-teams
while Stop=false
    if length (L) >1
  1.  Select a leader l, L= L/{l}, B = B +1
  2.  Define the max number of user in the MU-team N_max
      while i< N_max and k≤ length (L)
            if | V_k –V_l|< threshold
                 Add the mobile terminal to the MU-team;
                L= L/{k};
                 i=i+1;
            end if
            k=k+1;
       end while
      if i< N_max
          Stop=true;
       end if
    else
        Stop=true;
    end if
end while
# manage created MU-teams
for b=1 to B
    for k=1to N #N is the number of user in the MU-team
      if | V_k –V_l|> threshold
            user k will be removed from the MU-team
       end if
    end for
    while i< N_max and k≤ length (L)
          if | V_k –V_l|< threshold
              if RR_k < R_av
                   Add the mobile terminal to the MU-team.
                R_av = R_av- RR_k
                   L= L/{k}
                   i=i+1
          end if
        end if
            k=k+1
  end while
      Liberate unused resources
end for
```

The radio resources allocation for each MU-team will be communicated with the eNodeB by the MU-team leader only, which will decrease signalization traffic in the cell.

2.3 Optimization of PRB Allocation

In the network side, we work on the selection of the best radio resources to be allocated to each M-UE team. Therefore, the first issue is the estimation of the number of required PRBs by all users in the M-UE team, according to the required QoS for each user in the MU-team, which depends on the type of traffic. In our work, we considered both real-time and non-real-time traffic. For users with real-time applications, fixed rate are required and for those with non-real-time services, only minimum rate requirements are demanded. The required number of PRBs for user i is calculated as follow:

$$n_i = \left\lceil \frac{RR_i}{R_{PRB}} \right\rceil \tag{7}$$

With $[a]$ denotes the closed integer to a; RRi is the required rate of user i and RPRB denotes the peak capacity of one PRB. Assume 64 QAM modulation without coding, over 2 time slot (1 ms) a single PRB has 12 subcarriers and 14 symbols, or $12 \times 14 = 168$ resource elements (REs). Some of those REs are occupied by the PDCCH and the downlink reference signals, leaving about 120 REs per PRB to carry data on the downlink. In addition, with 64 QAM each RE holds 6 data bits, so the maximum data rate delivered by one PRB is equal to:

$$R_{PRB} = M_t \times 720\,\mathrm{Kb/s} \tag{8}$$

So, the total number of required PRBs of the users' team k is equal to:

$$N_k = \sum_i n_i \tag{9}$$

Next, the available band will be divided into sub-bands, each one with a number of successive PRBs equal to the estimated number of required PRBs. Our goal is to maximize the cell capacity.

Mathematically, we can present this maximization problem as:

$$\begin{aligned} &\text{maximize} C_T \\ &\textit{subject to}: \\ &\mathrm{C}_1: \sum_{\substack{j=1 \\ j\neq k}}^{K} \sum_{i=1}^{E} \left| Z_k H_{k,i}^{p} W_{j,i}^{p} \right|^2 \leq \gamma,\ \forall i \in \{1..E\}, j \in \{1..K\} \\ &\mathrm{C}_2: R_{k,i}^{RT} = \gamma_{RT},\ \forall k \in \{1..K^{RT}\} \\ &\mathrm{C}_3: R_{k,i}^{NRT} \geq \gamma_{NRT},\ \forall k \in \{1..K^{NRT}\} \\ &\mathrm{C}_4: \mathrm{C}_T \leq C_{threshold} \end{aligned} \tag{10}$$

The problem (10), described above, is resolved when performing the PRB allocation algorithm presented next.

Algorithm 2. PRBs Allocation.

Result: obtain the group of PRBs G_b^* to be allocated to each MU-team b.
Input:
Available bandwidth
MU-teams list with leader localization and QoS requirements for each MU-team.
Begin
For each MU-team b
1. Define Set of users with real time services
Define Set of users with non-real time services
2. Initialize G_b *
3. Resolve the problem (10)
End for
Update the available bandwidth to B= B- G_b*
If B <=threshold
Execute the MIH collaborated cell
End
End

The selected GPRB will be allocated to the correspondent MU-team. In order to maximize system throughput, we adopt downlink beamforming vector for each GPRB and also for each MU-team.

2.4 Congestion Prevention

Congestion control challenge is considered in our contribution by proposing a congestion prevention approach. The proposed approach aims to extend the system capacity by collaborating with neighors Cells/RATs, or by trigerring an efficient handover sessions. These two proposed schemes are detailed below.

Inter Cell and inter-RAT Collaboration
This step is executed by the MIIS server, which aims to form a group of cooperative heterogeneous cells to extend the available capacity. Heterogeneity here describes not only macro or small cells, but also different radio access technologies deployed by the operator in the area. This explains our choice on using MIH technology to ensure a simple communication between heterogeneous network equipment.

The selection of a collaborative cell or collaborative RAT is mainly based on the load of each. This step is executed when one of the LTE cells risks depleting its available resources, that can lead to a congested cell. In order to prevent such scenario, the MIIS is charged to collect information about all neighbors of the current cell in a limited area. Then, it communicates the list of candidates to the eNodeB. If the eNodeB finds LTE-A cell among the list of candidates, it will be selected to establish a collaboration through direct communication between eNodeBs via X2 interface. Otherwise, one of the least loaded RATs takes the place, and the two devices will exchange direct messages; thanks to the MIHF sub layer.

The selected cell/RAT is called “the grandmother cell/RAT”, because all new connections, and if it is necessary some MU-teams, will be served by the selected cell/RAT through their mother cell. In such case, the mother cell is considered like a remote node when serving new calls and handover request. As soon as possible, the mother eNodeB interrupts the connection with the “grandmother cell/RAT” and continues by itself to serve all connected user. Following, is the detailed the algorithm for the inter-cell/RAT collaboration establishment, which is executed by the eNodeBs.

Algorithm 3. Cloud MIR-RR.

Initialization:
e_c: Current eNodeB
C_{ec} : available capacity in the current cell
$Cloud_{RR}$: the RR cloud
C_{cloud}: total available capacity of the cloud
$Cloud_{RR}=\{e_c\}$
$C_{cloud}=C_{ec}$
Stop=false
While C_{cloud} < threshold and Stop=false
Sending a request to MIIS for searching a cooperative cells

- ♦ Neighbors’ capacity state request
- ♦ Select under loaded LTE cells / RATs
- ♦ Response by the list of candidate L_c sorted in ascending by load
  If L_c not empty
  - Select the first cell/RAT in the list
  - Establish collaboration via X2 or MIH.
  - C= C + available capacity in the collaborative cell
  - Cloud MIH = Cloud MIH □ {selected cell}

Else
Stop=true
end
End
If stop=true

Reject new call
Decrease the rate for some users
End
End

We define the new MIH primitive of service “MIH-Cooperate.req” exchanged between the MIH information server and the eNodeBs to ensure the collaborative resources allocation between LTE-A macro-cells.

MIH-based Handover Triggering
If there are no collaborative cells, the solution to prevent congestion state in the current cell is to block all new sessions and if necessary handover sessions will be triggered for

some users. The handover is assisted by the MIIS entity. The key step in the handover process is the target selection. Next, we describe the proposed target selection scheme, which aims to keep balanced load between neighbors while maintaining stability.

In order to make the target selection, we apply the WRMA (Weighted Rating of Multiple Attributes) approach, presented in [16]. The WRMA is designed to make handover decision and selection based on multiple network attributes reflecting the QoS. WRMA approach takes two major steps:

- Assigning values for network parameters in this first phase according to the traffic type and application priority. Authors in [16] used five parameters: line cost, capacity, delay, jitter and error rate. In our contribution, we consider in addition to these parameters the traffic load factor. We briefly describe these below:

– Line cost: The cost that may entail for using the service of the candidate base station, with a predetermined rate;
– Capacity: This is the remaining or available capacity of the candidate.
– Delay: This is the delay in packet transmission that a base station registers with both its mobile nodes and its CN. This is a way to look at the delay time that a mobile node records in transmitting data to the base station as well as the delay time the base station records in sending data to the CN via the backbone network.
– Jitter: This is the delay variation in transmitting data packets that a base station registers with both its mobile nodes and its CN.
– Error rate: It stands for degree of errors in packet transmission when a base station registers with its mobile nodes.
– Traffic load factor: It defines the threshold rate of each traffic type; this value is between zero and one since it is the ratio of data rate for one traffic type R_{TTi} by the total data rate in the cell R_T:

$$\rho_{TTi} = \frac{R_{TTi}}{R_T} \tag{11}$$

The traffic load factor is used to ensure load balancing between neighbors when selecting the target cell.

The WRAM scheme fixes four traffic types and eight levels of application priority. Subsequently, parameters take different values for each traffic type as presented in the Table 1.

Table 1. Assigned weight value of network parameters.

Parameter traffic type	Capacity	Delay	Jitter	Error	Cost	Traffic type load factor (ρ_{TT})
T1: Voice	3	6	5	2	6	3
T2: Video	5	6	6	3	3	5
T3: Best effort	3	1	1	8	1	3
T4: Background	2	1	1	1	1	1

We adopt the same logic used by the work in [18] to assign weight values for the load factor of each traffic type as new parameter. Indeed, first traffic type is voice packet which is not resource demanding and even if it presents the majority of traffic in the cell, there is no matter to serve new voice sessions in terms of system stability and load balancing between neighbor cells. So, we assign weight 3, meaning being relatively less important. The same situation for traffic type T3 and T4, to which we assign weight 3 and 1, respectively. Conversely, video traffic needs more resources and new sessions may tip the balance. Hence, in order to give more importance later in the decision we assign 5 for it.

Then, the WRAMA weighted ratios for each traffic type are calculated as below:

$$w_i = \frac{x_i}{\sum_{i=1}^{6} x_i} \tag{12}$$

Where x_i is the weight value assigned to parameter i

- When finishing with WRMA steps, the TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) [17, 18] method is used for selecting target network based on the obtained weight ratios of network attributes. The TOPSIS method computes a score for every candidate network according to parameters values that are readily available via a local MIH. So the candidate that scored the highest marks among all candidate cells will be selected.

Our approach occurs also in TOPSIS phase, in which we add the cell load factor (ρ) as a 7th attribute, since this parameter is independent from the application priority and the traffic type in the WRMA results. Hence, cell load will keep the same value for all traffic type.

The load factor for cell_i or RAT_i is calculated, according to the total number of resources in $\boldsymbol{\rho_T(t)}$, and the number of used ones $\boldsymbol{\rho_U(t)}$ for a period of time t, as shown below:

$$\rho_i(t) = \rho_U(t)/\rho_T(t) \tag{13}$$

The TOPSIS method takes place in five steps as described below:

Step 1: Parameters Normalization for All Candidates to Calculate TOPSIS Weight

TOPSIS weighted values are calculated from measurements presented in Table 2 using formula (14), which is presented below:

$$w_{ij} = \frac{p_{ij}}{\sum_{i=1}^{n} p_{ij}; j = 1..6} \tag{14}$$

Where, wij is the normalized TOPSIS weight and pij is the measured parameter of the candidate cell. Here, i and j are index for line and column of Table 2, respectively.

Table 2. Candidates cells parameters.

Parameter CCell	Capacity	Delay	Jitter	Error	Cost	Traffic_type load factor (ρ_{TT})	Cell/RAN Load factor (ρ)
C1	30	50	10	0.01	10	0.4	0.7
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Cn	40	30	20	0.12	6	0.01	0.3

Step 2: WRMA_RATIO Adjustment

In this step, TOPSIS normalized values will be adjusted using WRMA weight values, that result from the first phase of the approach as detailed above. This task depends to the traffic type of the current mobile node session and its application priority level. It consists simply of multiplying parameters values for each candidate cell. This step does not include the load factor parameter since it is not used in the first phase.

Step 3: Ideal Solution and Negative-ideal Solution Estimation

Based on values resulting from last step the ideal solution S* takes the best value of each parameter for all candidate cells and the worst value will be assigned to the negative-ideal solution S- according to the parameter itself. Indeed, if the highest value is the best, the smallest value will be the worst, which is the case of capacity parameter, but for others that will be the inversed.

Mathematically, the ideal solution is modeled as below:

$$\begin{aligned} S^* &= \left[p_1^* p_2^* p_3^* p_4^* p_5^* p_6^*\right] \\ &= \left\{\left(\max p_{i1} | i \in \{i..n\}\right), \left(\min p_{ij} | i \in \{i..n\}, j \in \{2..6\}\right)\right\} \end{aligned} \tag{15}$$

Formula (15) is used to find the negative-ideal solution definition:

$$\begin{aligned} S^- &= \left[p_1^- p_2^- p_3^- p_4^- p_5^- p_6^-\right] \\ &= \left\{\left(\max p_{i1} | i \in \{i..n\}\right), \left(\min p_{ij} | i \in \{i..n\}, j \in \{2..6\}\right)\right\} \end{aligned} \tag{16}$$

Step 4: Distance of each CRAN among the Ideal Solution and Negative-Ideal Solution

In order to calculate the score for candidate cell, values of the ideal solution and similarly the negative-ideal solution are weighted for all candidate cell using formula (17) below:

$$\begin{aligned} s_i^* &= \sqrt{\sum_{j=i}^{n} \left(p_{ij} - p_j^*\right)^2} \\ s_i^- &= \sqrt{\sum_{j=i}^{n} \left(p_{ij} - p_j^-\right)^2} \; i = 1..n; j = 1..6 \end{aligned} \tag{17}$$

Step 5: Candidate Cells Scores Estimation
This final step is to calculate the weighted sum of each candidate cell using the following formula:

$$\begin{aligned} c_i^* &= s_i^- / \left(s_i^- + s_i^*\right) \\ C^* &= \left[c_i^*\right]; i = 1..n \end{aligned} \tag{18}$$

Finally, the candidate with the highest score will be selected as target.

2.5 Radio Resources Allocation Track

To ensure the effectiveness of the radio resources allocation approach, a final step of control is needed. This step consists of allocating track to see if there is over served MU-teams of users. In such case:

- Withdraw unused resources to be reallocated to underserved MU-teams or to incoming call/user;
- Or see the possibility to include new users/calls to one of the existing MU-teams according to the proposed scheme described above.

Also, our solution takes into account resources as soon as it becomes available after a terminal leaving.

3 Performance Evaluation

This section illustrates simulation results to evaluate our proposed algorithms in terms of system throughput. We worked with the Matlab-based LTE-A System Level simulator developed by the TU Wien Telecommunications Institute [15, 19].

3.1 Simulation Parameters and Scenario

The adopted system parameters in simulations are presented in Table 3, which is given below.

Table 3. Simulation parameter.

Parameter	Description
Duration of simulation	200 TTIs
Number of users per macro cell	100 UEs/MC; 591 in total
M_r	4
M_t	2
Cell radius	250 m
System bandwidth	20 MHz
Number of PRBs	100 RBs
PRB bandwidth	180 kHz
Real time requirement [14]	384 kbps
Non real time requirement [14]	32 kbps

Our approach aims to maximize cell capacity. To highlight this issue, we studied the total throughput that corresponds to the sum of the effective throughput of all scheduled users at each time slot. Since small cells deployment is expected to be dense in future LTE-A system, we felt that it would be interesting to consider such scenario in the simulation model. Indeed, the main objective of this technology is to extend system capacity, especially for edge users, but in the other side inter small-cells interference will increase dramatically and put more pressure on the macro cells users. Thus, in simulations we considered femtocell deployment scenario, which is provided by the simulator.

3.2 Simulation Results and Discussion

In this subsection, we present the simulation results carried out in order to evaluate the performance of the proposed Radio Resources Allocation (RRA) scheme and compare it with other competing schemes reported in the literature.

In this work, we compare results of two different configurations:

- Configuration 1: An interfered system and no optimization is performed with femtocells deployment only;
- Configuration 2: An interfered system with joint MUE-teams and femtocells deployment.

We also adopt as performance evaluation metrics the wideband SINR as well as the average throughput that corresponds to the average effective throughput of scheduled users at each time slot. The simulator provides the Empirical Cumulative Distribution Function (ECDF) statistics of these results, which makes it easier to analyze the achieved results.

The first result is shown in Fig. 3, which depicts the ECDF for all users' throughput to compare performance when considering macro-cells and femtocells with and without the integration of our optimization algorithms. It can be seen that the analytical results provided by our solution outperforms the nominal case. However, with our scheme, the interference is minimized; providing better performance.

We observe that MUE-teams' deployment besides femtocells, offers a higher average throughput in comparison to the initial configuration. Indeed, the maximum achieved throughput when deactivating the proposed optimizations algorithms is about 9.4 Mb/s. While, when performing our algorithms, 20% of users exceed this value with the possibility to achieve 11.8 Mb/s.

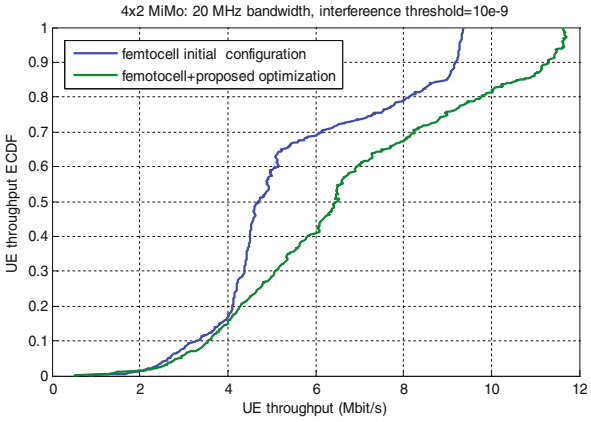


Fig. 3. Empirical CDF of global throughput.

Figure 4 shows performance of scheduling SINR in form of the empirical Cumulative Distribution Function (CDF) when performing the proposed scheme. By examining the curve, we conclude that for 90% of users, our scheme is able to find a better choice of resources allocation than with the initial configuration case. We evaluate the total average throughput according to the MU-team deployment.

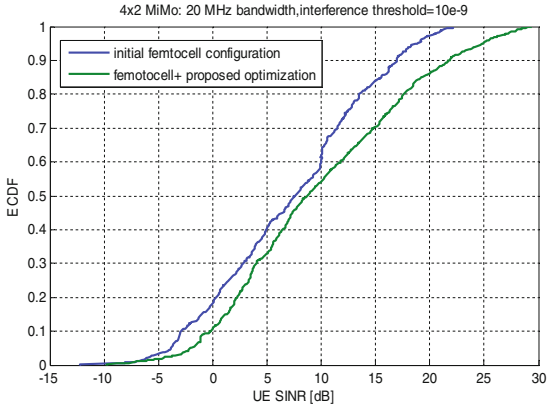


Fig. 4. Wideband SINR.

Figure 5 shows the variation of the average instantaneous throughput in the macro cell with the concentrations of femtocells when performing our proposed schemes. Even, more femtocells means more interference inside the macro cell where users may suffer strong interference due to the transmission of small cells. Consequently, macro users' throughput may be affected dramatically. According to achieved result, despite a significant increase of femtocells number, we can observe that macro users throughout show a slower slight decrease thanks to the interference limitation. We can conclude

that our optimization solution maintains the UE throughput level in various densities of small cells. Furthermore, our approach ensures cooperation between scheduled users as well as between eNodeBs. Such cooperative communication can reduce the transmitted power of macro eNodeBs at MUE-teams leaders, which results in higher channel capacity for users.

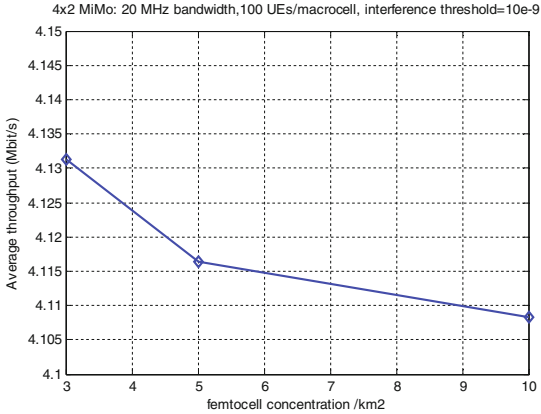


Fig. 5. Femtocells concentration deployment effect on average throughput.

In addition, we have studied the effect of the interference threshold value on resource allocations in Fig. 6, where we have varied the threshold. As expected, when the interference threshold increases, the achieved data rates for macro users are affected. However, we can observe that the total MU-team throughput shows a slight decrease according to the increase of interference level.

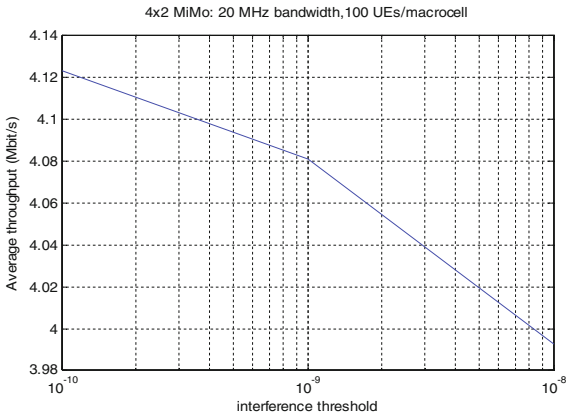


Fig. 6. Effect of interference threshold level on average MU-teams throughput.

4 Conclusion

The LTE-A small cells deployment is expected to be more and denser. This generates potential inter-small cells interference that affect macro cell UE, contrarily to indoor small cells UE users that will benefit of high quality links. Classical radio resource allocation and interference mitigation techniques cannot address the challenge of limiting interference between neighboring small cells and maintaining a high level of reliability for macro UE communications. Moreover, we have proposed a RRA approach, which can be implemented in stand-alone as well as networked small cells scenarios.

We have proposed a cooperative RR allocation approach for LTE-A system. We have devised a new way to manage active mobile users in the macro cell, in order to minimize inter-user and inter-small cells interference and to improve their QoS. Furthermore, we have considered the multi-RAT aspect of LTE-A system; we also motivate the importance of inter-cell and inter-RAT cooperation. The proposed solution makes the heterogeneity an enabler to improve system capacity. We further propose a novel PRBs selection and allocation that optimize resources exploitation. Also in case of congestion state handover session will be triggered and a modified WRMA approach is proposed to select the target while considering candidates state and load balancing between them. In the end of the article, we have discussed the effectiveness of the proposed approach in a two-tier network in terms of system capacity, data rate loss, and performance at both macro UEs and small cells UEs. Simulation results have shown that the proposed scheme maximizes the system throughput while guaranteeing QoS for users.

References

1. Hossain, E., Kim, D.I., Bhargava, V.K.: Cooperative Cellular Wireless Networks. Cambridge University Press, Cambridge (2011)
2. Papathanasiou, C., Dimitriou, N., Tassiulas, L.: Dynamic radio resource and interference management for MIMO-OFDMA mobile broadband wireless access systems. Comput. Netw. **57**(1), 3–16 (2013)
3. Alavi, S.M., Zhou, C., Gen, W.W.: Efficient resource allocation algorithm for OFDMA systems with delay constraint. Comput. Commun. **36**(4), 421–430 (2013)
4. Miyazaki, N., Wang, X., Fushiki, M., Akimoto, Y., Konishi, S.: A proposal for radio resource allocation of TDM inter-cell interference coordination to heterogeneous networks with pico cells in LTE-advanced. In: 76th IEEE Vehicular Technology Conference (VTC Fall), pp. 1–5, Quebec, Canada, September 2012
5. Tang, J., Daniel, K.C.S., Alsusa, E., Hamdi, K.A., Shojaeifard, A.: Resource allocation for energy efficiency optimization in heterogeneous networks. IEEE J. Sel. Areas Commun. **33** (10), 2104–2117 (2015)
6. Akkarajitsakul, K., Hossain, E., Niyato, D., Kim, D.I.: Game theoretic approaches for multiple access in wireless networks: a survey. IEEE Commun. Surv. Tutorials **13**(3), 372–395 (2011)

7. Alavi, S.M., Zhou, C., Gen, W.W.: Distributed resource allocation scheme for multicell OFDMA networks based on combinatorial auction. In: 76th IEEE Vehicular Technology Conference (VTC Fall), Quebec, Canada, pp. 1–5, September 2012
8. Lee, D., Seo, H., Clerckx, B., Hardouin, E., Mazzarese, D., Nagata, S., Sayana, K.: Coordinated multipoint transmission and reception in LTE-advanced: deployment scenarios and operational challenges. IEEE Commun. Mag. **50**(2), 148–155 (2012)
9. Sawahashi, M., Kishiyama, Y., Morimoto, A., Nishikawa, D., Tanno, M.: Coordinated multipoint transmission/reception techniques for LTE-Advanced [coordinated and distributed MIMO]. IEEE Wirel. IEEE Commun. Mag. **17**(3), 26–34 (2010)
10. Irmer, R., Droste, H., Marsch, P., Grieger, M., Fettweis, G., Brueck, S., Mayer, H.P., Thiele, L., Jungnickel, V.: Coordinated multipoint: Concepts, performance and field trial results. IEEE Commun. Mag. **49**(2), 102–111 (2011)
11. Somekh, O., Simeone, O., Bar-Ness, Y., Haimovich, A.M., Shamai, S.: Cooperative multicell zero-forcing beamforming in cellular downlink channels. IEEE Trans. Inf. Theory **55**(7), 3206–3219 (2009)
12. Huh, H., Tulino, A.M., Caire, G.: Network MIMO with linear zero-forcing beamforming: large system analysis, impact of channel estimation, and reduced-complexity scheduling. IEEE Trans. Inf. Theory **58**(5), 2911–2934 (2012)
13. Lee, G., Park, J., Sung, Y., Yukawa, M.: Coordinated beamforming with relaxed zero forcing. In: International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, pp. 1–5, November 2011
14. Rysavy, P.: EDGE, HSPA, LTE: The Mobile Broadband Advantage. http://www.rysavy.com/Articles/2007_09_Rysavy_3GAmericas.pdf
15. Mehlfuhrer, C., Ikuno, J.C., Šimko, M., Schwarz, S., Wrulich, M., Rupp, M.: The Vienna LTE simulators - enabling re-producibility in wireless communications research. EURASIP J. Adv. Sig. Process. (2011)
16. Shin-Jer, Y., Wen-Chieh, T.: Design novel weighted rating of multiple attributes scheme to enhance handoff efficiency in heterogeneous wireless networks. Comput. Commun. **36**(14), 1498–1514 (2013)
17. Behzadian, M., Otaghsara, S.K., Yazdani, M., Ignatius, J.: A state-of the-art survey of TOPSIS applications. Exp. Syst. Appl. **39**(17), 13051–13069 (2012)
18. Paul Yoon, K., Ching-Lai, H.: Multiple Attribute Decision Making: An Introduction, vol. 104. Sage Publications, Thousand Oaks (1995)
19. Ikuno, J.C., Wrulich, M., Rupp, M.: System level simulation of LTE networks. In: IEEE Vehicular Technology Conference, VTC 2010-Spring, Taipei, Taiwan, pp. 1–5, May 2010

Author Index

M.S. Obaidat et al. (eds.), *Simulation and Modeling Methodologies, Technologies and Applications*, Advances in Intelligent Systems and Computing 676,
https://doi.org/10.1007/978-3-319-69832-8