MODELLING AND SIMULATION 2018

THE EUROPEAN SIMULATION

AND

MODELLING CONFERENCE

2018

ESM®'2018

EDITED BY

Dieter Claeys

and

Veronique Limère

OCTOBER 24-26, 2018

GHENT

BELGIUM

A Publication of EUROSIS-ETI

The 32nd Annual European Simulation and Modelling Conference 2018

GHENT, BELGIUM

OCTOBER 24-26, 2018

Organised by

ETI - The European Technology Institute

Sponsored by

EUROSIS - The European Simulation Society

Ghent University

Co-Sponsored by

University of Skovde

University of Žilina

GODAN

Hosted by

NH Gent Belfort
Ghent, Belgium

EXECUTIVE EDITOR

PHILIPPE GERIL
(BELGIUM)

EDITORS

General Conference Chairs

Dieter Claeys
Ghent University
Department of Industrial Systems Engineering and Product Design
Zwijnaarde (Gent) and Flanders Make, Belgium
Veronique Limère
Ghent University
Department of Business Informatics and Operations Management
Ghent, Belgium

Past Conference Chair
Paulo J.S. Gonçalves
Instituto Politécnico de Castelo Branco, Portugal,
and
IDMEC/LAETA, Instituto Superior Técnico, Universidade de Lisboa, Portugal

Journal Publication Chairs
Yan Luo, NIST, Gaithersburg, USA
Peter Lawrence, Swinburne University, Australia
Dr. Wan, International Islamic University Malaysia

ESM Conference Chair
António Carvalho Brito, FEUP - University of Porto, Porto, Portugal

INTERNATIONAL PROGRAMME COMMITTEE

Methodology and Tools
Claudia Krull, Otto-von-Guericke University, Magdeburg, Germany
J. Manuel Feliz Teixeira, University of Porto, Porto, Portugal
Jerzy Respondek, Silesian University of Technology, Gliwice, Poland

Discrete Simulation Modeling Techniques and Tools
Renato Natal Jorge, FEUP - University of Porto, Porto, Portugal
Helge Hagenauer, Universitaet Salzburg, Salzburg, Austria
Sophie Hennequin, ENIM, Metz Cedex, France
Nuno Melão, Instituto Politécnico de Viseu, Viseu, Portugal

Queueing Models
Ahmad Al Hanbali, King Fahd University of Petroleum & Minerals - KFUPM, Saudi Arabia
Esa Hyytiä, University of Iceland, Reykjavik, Iceland
Ioannis Dimitriou, University of Patras, Patras, Greece
Jan-Pieter Dorsman, University of Amsterdam, Amsterdam, The Netherlands
Stijn De Vuyst, Ghent University, Ghent, Belgium

Simulation and Artificial Intelligence
Helder Coelho, Fac Ciencias, Lisbon, Portugal
Paulo Cortez, University of Minho, Guimareas, Portugal
Martin Hruby, Brno University of Technology, Brno, Czech Republic
Vladimir Janousek, Brno University of Technology, Brno, Czech Republic
Leon Rothkrantz, Prague University of Technology, Prague, Czech Republic

INTERNATIONAL PROGRAMME COMMITTEE

Agent Based Simulation
Zisheng Huang, Vrije Universiteit Amsterdam, The Netherlands
Ioan Alfred Letia, TU Cluj Napoca, Romania
Isabel Praca, Ist. Superior do Porto, Porto, Portugal

Simulation and Optimization
José António Oliveira, Universidade do Minho, Campus de Gualtar, Braga, Portugal
Janos-Sebestyen Janosy, Hungarian Academy of Sciences, Budapest, Hungary

High Performance Large Scale and Hybrid Computing
Jan Broeckhove, University of Antwerp, Antwerp, Belgium
Pierre Siron, ONERA, Toulouse, France
Jingjing Wang, SUNY Binghamton University, New York, USA

Simulation in Education and Graphics Visualization
Ana Luísa Ramos, University of Aveiro, Aveiro, Portugal
Ranjit Singh, UB Patient Safety Research Center, University of Buffalo, Buffalo NY USA

Simulation in Environment, Ecology, Biology and Medicine
Joel Colloc, Université du Havre, Le Havre, France
Laurent Perochon, VetaGro-Sup, Lempdes, France
Filipe Pinto, Polytechnic Institute of Leiria, Portugal

Analytical and Numerical Modelling Techniques
Ana M. Camacho, UNED, Madrid, Spain
Clemens Heitzinger, TU Vienna, Vienna, Austria

Web and Cloud Based Simulation
Manuel Alfonseca, Universidad Autonoma de Madrid, Spain
Yan Luo, NIST, Gaithersburg, USA
Jose Machado, University of Minho, Braga, Portugal

Cosmological Simulation
José Manuel Feliz-Teixeira, University of Porto, Porto, Portugal
Philippe Geril, ETI Bvba, Ostend, Belgium

Intelligent Systems
José Machado, Universidade do Minho, Braga, Portugal
Manuel Filipe Santos, Universidade do Minho, Guimarães, Portugal

Simulation with Petri Nets
Francois Siewe, De Montfort University, Leicester, United Kingdom
Pascal Berruet, Universite Bretagne Sud, Lorient, France
Stefano Marrone, Seconda Universita degli Studi di Napoli, Naples, Italy
Alexandre Nketsa, LAAS-CNRS, Toulouse, France

Bond Graphs Simulation
Jesus Felez, Univ. Politecnica de Madrid, Spain
Andre Tavernier, BioSim, Brussels, Belgium

DEVS
Fernando Tricas, Universidad de Zaragoza, Spain

Fluid Flow Simulation
H.A.Nour Eldin, University of Wuppertal, Germany
Markus Fiedler, Blekinge Institute of Technology, Sweden

INTERNATIONAL PROGRAMME COMMITTEE

Emergency Risk Management Simulation
Véronique Baudin, LAAS-CNRS, Université de Toulouse, Toulouse, France
Konstantinos Kirytopoulos, University of South Australia, Australia
Lode Vermeersch, Delcredere, Brussels, Belgium

EUROPEAN SIMULATION AND MODELLING CONFERENCE 2018

All articles published in these Proceedings have been peer reviewed.

EUROSIS-ETI Publications are ISI-Thomson and IET referenced.
ESM® Proceedings are indexed on ISI-Thomson, IET, SCOPUS and Elsevier Engineering Village.

A CIP Catalogue record for this book is available from the Royal Library of Belgium under nr.12620

For permission to publish a complete paper write EUROSIS, c/o Philippe Geril, ETI Executive Director, Greenbridge Science Park, Ghent University – Ostend Campus, Wetenschapspark 1, Plassendale 1, B-8400 Ostend, Belgium.

EUROSIS is a Division of ETI Bvba, The European Technology Institute, Torhoutsesteenweg 162, Box 4.02, B-8400 Ostend, Belgium

Printed in Belgium by Reproduct NV, Ghent, Belgium
Cover Design by C-Design Bvba, Wielsbeke, Belgium

EUROSIS-ETI Publication

ISBN: 978-9492859-05-1
EAN: 9789492859051

Preface

Dear participants,

It is our honour to welcome each of you to the 32nd European Simulation and Modelling Conference - ESM'2018 held in association with Ghent University. During two and a half days, this conference aims to be a meeting place where scientists and technicians in the field of Simulation and Modelling can find a discussion forum to promote research and exchange ideas. The programme is organized to allow for ample opportunity to meet and discuss, encouraging a high level of interaction and reflection on new research ideas.

We have some 42 participants from 20 countries with presentations grouped in 10 main themes such as Simulation Methodology and Tools, Simulation in Engineering Processes, Web and Cloud Based Simulation, Supply Chain Simulation, Simulation in Energy and Power Systems, and Queueing Models.

Furthermore, we are honoured to announce our keynote speakers, Ivo Adan (Technical University of Eindhoven), talking about multi-class queueing applications in automated warehousing systems, and Jan Van Impe (University of Leuven), who will talk about advanced dynamic optimization of (bio)chemical processes under parametric uncertainty. Moreover, we invited two additional speakers, Jacob Kegerreis (Durham University), talking about smoothed-particle hydrodynamics (SPH) simulations, and Leon Rothkrantz (Czech Technical University), talking about flooding disaster models. We are very grateful that they could accept our invitation to share their expertise with us.

We are indebted to a number of people without whom the conference would not be possible. First of all, we would like to thank all authors for sharing their contributions. Our thanks also go to all members of the International Program Committee for the critical reviewing work that was key to maintaining the high scientific quality of the conference.

Furthermore, we would in particular like to thank Philippe Geril from EUROSIS, whose continued dedication and hard work, as the conference organizer, was indispensable for a succesful organization of ESM'2018.

Finally, we would like to wish you all a stimulating and inspiring experience at the conference and a pleasant stay in the beautiful city of Ghent.

Veronique Limère
Dpt. Business Informatics and Operations Management
and
Dieter Claeys
Dpt. Industrial Systems Engineering and Product Design
Ghent University, Belgium

ESM'2018 General Chairs

CONTENTS

Preface ..... **IX**
Scientific Programme **1**
Author Listing **227**

OPTIMIZATION METHODOLOGY

Statistical Optimization with Averaging applied to a 125 Variable and 125 Nonlinear System of Equations
William Conley **5**

Statistical Optimization applied to a large Nonlinear System of Equations
William Conley **11**

On Parallel Conformant Planning as an Optimization Problem
Adam Galuszka, Tomasz Grzejszczak, Jaroslaw Smieja, Adrian Olczyk and Jerzy Kocerka **17**

Multi-Objective Optimization Driven Construction of Uniform Priors for Likelihood-Free Parameter Inference
Prashant Singh and Andreas Hellander **22**

A Simulation Based Technique for Continuous Space Embedding of Discrete Parameter Queueing Systems
Neha Karanjkar, Madhav Desai and Shalabh Bhatnagar **28**

MACHINE LEARNING AND AI

Application of Machine Learning to model a Biological Reactor in a Wastewater Treatment Plant
Bartosz Szeląg and Jan Studziński **39**

Modeling Emergence by Integrating DEVS and Machine Learning
Daniel Foguelman and Rodrigo Castro **44**

Hand Gesture Classification with Use of Convolutional Neural Networks
Tomasz Grzejszczak, Robert Roth and Reinhard Moller **49**

The Problem with Asynchronous Updating
Joseph Kehoe **54**

SIMULATION DRIVEN ENGINEERING

Model-Based Simulation of a Hydraulic Open Loop Rotary Transmission with Automatic Regulation of Hydraulic Motor (Part 1: Modelling)
Mait Harf and Gunnar Grossschmidt **63**

CONTENTS

Model-Based Simulation of a Hydraulic Open Loop Rotary Transmission with Automatic Regulation of Hydraulic Motor (PART 2: Simulation)
Mait Harf and Gunnar Grossschmidt **.....69**

PRODUCTION PLANNING

Coupling of Rigid Body Dynamics with Structural Mechanics to include Elastic Deformations in a Real-Time Capable Holistic Simulation for Digital Twins
Max Krause, Kai-Uwe Schröder, Dorit Kaufmann, Tobias Osterloh and Jürgen Rossmann **.....77**

Process-based Ship Production System Modeling for Supporting Decision Processes
Dongsu Jeong and Yonnho Seo **.....82**

Development, Implementation and Evaluation of a Complexity Measure for the Work of Assembly Teams in One-Piece-Flow Assembly Systems Employing Simulation Studies
Benedikt A. Latos, Peyman Kalantar, Florens L. Burgert, Matthias G. Arend, Verena Nitsch, Philipp M. Przybysz and Susanne Mütze-Niewöhner **.....88**

Integrated EPQ and Periodic Condition-Based Maintenance
Alp Darendeliler, Dieter Claeys, El-Houssaine Aghezzaf and Abdelhakim Khatab **.....95**

ASSEMBLY LINE OPTIMIZATION

Optimization of a Supply Chain Game Orders with Order Cost
Florian Boyrie, Nicolas Pech-Gourg and Thibaud Serieye **.....103**

The Influence of Line Balancing on Line Feeding for Mixed-Model Assembly Lines
Hendrik Wijnant, Nico André Schmid and Veronique Limère **.....106**

Implementation of a Cyber-Physical Systems Simulation Components Allocation Tool
Henrick Deschamps, Gerlando Cappello, Janette Cardoso and Pierre Siron **...112**

SUPPLY CHAIN ANALYSIS AND VEHICLE SCHEDULING

Variable and Class-Dependent Service Capacity with a Multi-Class Arrival Process
Jens Baetens, Bart Steyaert, Herwig Bruneel and Dieter Claeys **.....123**

CONTENTS

Simulation Analysis tackling Demand Stochasticity through Safety Stock and Lateral Transshipments
Benedikt De Vos and Birger Raa **.....131**

Bidirectional Service Strategy for Reusable Transportation Packaging (RTPs) Pooling System
Di Zhang and Uwe Clausen **.....139**

Integrated Electric Vehicle Scheduling and Charging Infrastructure Planning for Public Transport
Hubert Büchter and Sebastian Naumann**.....147**

SIMULATION IN ENERGY

An Offshore Energy Simulation through Flow Networks: *Cel* within the *MSP Challenge 2050* Simulation Game Platform
Kevin Hutchinson, Harald Warmelink, Wilco Boode, Carlos Santos and Igor Mayer**.....157**

Non Cooperative Game Theoretic Approach for Residential Energy Management in Smart Grid
Ilyes Naidji, Moncef Ben Smida, Mohamed Khalgui and Abdelmalik Bachir **.....164**

Power System Planning supported by Big Data
Samuel Marrero Vera, José Évora Gómez and José Juan Hernández Cabrera **.....171**

Simulation of the LB-LOCA in CANDU 6 Reactors with RELAP/SCDAPSIM(IUA). Modelling Effects of the Break Location
Roxana-Mihaela Nistor-Vlad, Daniel Dupleac, Ilie Prisecaru, Marina Perez-Ferragut and Chris Allison **.....177**

ELECTRONIC NETWORK SIMULATION

New Scheduling Mechanism in Multi-Channel Reconfigurable WSN under QoS and Energy Constraints
Yousra Ben Aissa, Olfa Mosbahi, Mohamed Khalgui and Abdelmalik Bachir **...187**

Multi-Agent Model of Infrastructural Return for an Intermediary Service Provider
Anton Ivaschenko and Stanislav Korchivoy **.....192**

CONTENTS

ENVIRONMENTAL SIMULATION

Stochastic Model of the Joint Time-Series of Air Temperature and Atmospheric Pressure
Nina A. Kargapolova **199**

Temporospatial Epidemic Simulations using Heterogeneous Computing
Dhananjai M. Rao **205**

SIMULATION IN ECONOMICS AND FINANCE

Volatility Regime Analysis of Bitcoin Price Dynamics using Markov Switching Garch Models
Anna Bonello and David Suda **213**

Models as Part of the Welfare System as Part of The Economic Measures for The State of Crisis in Building Resilience
Jozef Ristvej, Roman Ondrejka, Ladislav Šimák, Michal Titko and Michaela Jánošíková **219**

SCIENTIFIC PROGRAMME

OPTIMIZATION METHODOLOGY

STATISTICAL OPTIMIZATION WITH AVERAGING APPLIED TO A 125 VARIABLE AND 125 NONLINEAR SYSTEM OF EQUATIONS

William Conley
Austin E. Cofrin School of Business
480B Wood Hall
University of Wisconsin-Green Bay
Green Bay, Wisconsin 54311-7001
U.S.A.
Conleyw@uwgb.edu

KEY WORDS
Nonlinear systems, modal averaging, multi stage Monte Carlo optimization, statistical theory and application.

ABSTRACT

The fields of linear algebra and subsequent linear programming have made great strides in solving large scale systems of equations and linear optimization problems. However, so many times in the real worlds of big business, science and economics, the truly practical models are nonlinear. Therefore, with linear programing and linear algebra one obtains the exact solution to the wrong question. However, some of the linear models are accurate enough to be very useful. Presented here will be statistical optimization (or multi stage Monte Carlo optimization) simulation with modal averaging applied to a 125 variable system of 125 nonlinear equations and solved for the true optimal solution with no equation errors. Then a nonlinear system with no solution will be studied to see that even in that case a perhaps useful approximation solution can be produced with multi stage Monte Carlo optimization simulation.

INTRODUCTION

Consider the system of equations:

$x_1 + x_2 = 11$

$2x_1 + 7x_2 = 47$

It could be solved by substitution or cancellation. However, a third way would be to transform it into the optimization problem: minimize $f(x_1, x_2) = |x_1 + x_2 - 11| + |2x_1 + 7x_2 - 47|$ and write a computation simulation program of the multi stage Monte Carlo (MSMCO) type to cross the sampling distribution of the feasible solution space to the minimum error of zero. In this case that would occur at $x_1 = 6$ and $x_2 = 5$. Please see Figures 1 and 2 for partial geometric and statistical representation of MSMCO crossing feasible solution space to look for the minimum solution.

Now let us look at a large problem.

A 125 X 125 NONLINEAR SYSTEM

It is desired to find an all whole number solution in the range 1 to 50 for the following 125 equations with 125 variables system:

$x_i^4 + x_{126-i}x_{125-i} + x_{124-i} = c_i$ (i)

for i= 1, 2, 3, . . . 65

$x_{66}^4 + x_{71}x_{76} + x_{81} = 331{,}898$ (66)

and for i = 67, 68, . . . 122 the following

$x_i^4 + x_{126-i}x_{125-i} + x_{124-i} = 4{,}477{,}711$ (67)

and

c_i

and also

$x_{123}^4 + x_3x_2 + x_1 = 15{,}225$ (123)

$x_{124}^4 + x_{16}x_{51} + x_{89} = 161{,}367$ (124)

$x_{125}^4 + x_{27}x_{99} + x_{42} = 131{,}059$ (125)

$1 \le x_i \le 50$ and all whole numbers for the i=1, 2, 3, . . . 125 equations is desired. All of the constants c_i's for the values for the right hand sides of the equations follow here in the pattern:

c_1 c_2 c_3 c_4 c_5
c_6 c_7 c_8 c_9 c_{10}
c_{121} c_{122} c_{123} c_{124} c_{125}

Note that the five written out above are underlined for ease of reading and the L_js are the left hand side of the equation.

707672	50856	1874312	2560355	7336
29206	5308950	51337	280941	1691
3418939	6250098	130528	1186414	1048772
6250053	234329	3113057	3772	532198
369	614888	6250296	5309455	2206
5766822	105855	1049425	1187136	39793
4101888	4101927	390947	290	443
132	391447	2315331	8602	40488
337	105756	6251138	2086785	1051053
1598	609	1874454	2085301	6250377
532236	1136	55	194525	436
1875567	4100911	3111996	1735	279992

2085200	390804	234844	3374	2175
331898	4477711	3223	2561668	1500757
6626	48	810037	83706	4102013
6033	52034	130436	531485	5764967
6251266	1050494	1337243	234396	2825868
6250165	560376	4478432	83550	632
1336373	1416	5309586	533480	4101288
810494	2314515	195106	2560884	6250146
3418947	282269	16043	130404	105037
2825992	1186237	2560974	1133	30194
51700	1049253	2289	16797	21017
1048729	1336729	160733	391258	390782
810397	16136	15225	161367	131059

Therefore, the 125 equations with 125 variables system is transformed to minimize $f(x_1,x_2,x_3, \ldots x_{125}) = \sum_{j=1}^{125} |L_j-C_j|$

for $1 \le x_i \le 50$ and i=1, 2, . . . 125. All x_i's are whole numbers and L_j and C_j are the left and right hand sides respectively of the 125 equations. Please note that this simulation is started with real values for the x_i's. However, once the modal averaging starts the whole numbers start to appear on the way to the exact solution of all whole numbers by the end of the simulation optimization.

THE SOLUTION ALGORITHM

The solution plan is briefly illustrated in Figure 1. There are $50^{125} \approx 2.3x10^{212}$ feasible solutions. Therefore, any type of complete search would take forever.

The solution approach used here (the algorithm) was to do two multi stage Monte Carlo optimization (MSMCO) simulation tries of 24 stages each in the x_i = 1 to 50 region for i = 1, 2, . . . 125 drawing 60,000 sample feasible solutions at each stage of the ever free floating always moving (and decreasing in size) geometric shapes that are chasing across the feasible solution space. These first attempts are done with real arithmetic.

However, then these two approximate answers are compared coordinate-wise and if any of the variables values are within .015 of each other, their average is rounded to the nearest whole number and pinned down at that value for the rest of the simulation. After these two solution attempts two more are tried (numbers 3 and 4) with the 1 and 2 tries common values still pinned down. Then the comparison of these two 60,000x24 simulation solution attempts is compared for each pair and again if their difference is ≤ .015 it is averaged and pinned down (rounded to the nearest whole number) for the rest of the simulation. These pairs of 60,000x24 simulations are done six times with subsequent pinning down taking place as the whole solution then emerges on the 13th pass (simulation). It is presented here with no error in all 125 equations.

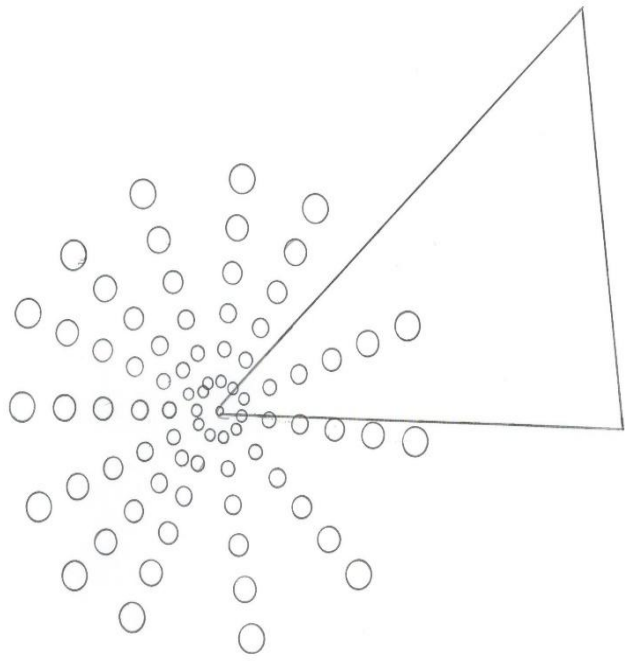


Figure 1: Twelve Solution Tries were Averaged to Lead to the Solution on the Thirteenth Attempt

These 13 solution attempts of 60,000x24x13 = 18,720,000 feasible solutions were all controlled by an outer loop in the program so that this program ran in 4 minutes of computer time in the author's office and printed the solution:

X_1=29	X_2=15	X_3=37	X_4=40	X_5=9
X_6=13	X_7=48	X_8=15	X_9=23	X_{10}=6
X_{11}=43	X_{12}=50	X_{13}=19	X_{14}=33	X_{15}=32
X_{16}=50	X_{17}=22	X_{18}=42	X_{19}=7	X_{20}=27
X_{21}=2	X_{22}=28	X_{23}=50	X_{24}=48	X_{25}=2
X_{26}=49	X_{27}=18	X_{28}=32	X_{29}=33	X_{30}=14
X_{31}=45	X_{32}=45	X_{33}=25	X_{34}=3	X_{35}=4
X_{36}=1	X_{37}=25	X_{38}=39	X_{39}=9	X_{40}=14
X_{41}=7	X_{42}=18	X_{43}=50	X_{44}=38	X_{45}=32
X_{46}=4	X_{47}=3	X_{48}=37	X_{49}=38	X_{50}=50
X_{51}=27	X_{52}=5	X_{53}=2	X_{54}=21	X_{55}=3
X_{56}=37	X_{57}=45	X_{58}=42	X_{59}=5	X_{60}=23
X_{61}=38	X_{62}=25	X_{63}=22	X_{64}=7	X_{65}=6
X_{66}=24	X_{67}=46	X_{68}=6	X_{69}=40	X_{70}=35
X_{71}=9	X_{72}=1	X_{73}=30	X_{74}=17	X_{75}=45
X_{76}=8	X_{77}=15	X_{78}=19	X_{79}=27	X_{80}=49
X_{81}=50	X_{82}=32	X_{83}=34	X_{84}=22	X_{85}=41
X_{86}=50	X_{87}=40	X_{88}=46	X_{89}=17	X_{90}=5
X_{91}=34	X_{92}=6	X_{93}=48	X_{94}=27	X_{95}=45
X_{96}=30	X_{97}=39	X_{98}=21	X_{99}=40	X_{100}=50
X_{101}=43	X_{102}=23	X_{103}=11	X_{104}=19	X_{105}=18
X_{106}=41	X_{107}=33	X_{108}=40	X_{109}=1	X_{110}=13

X_{111}=15	X_{112}=32	X_{113}=6	X_{114}=11	X_{115}=12
X_{116}=32	X_{117}=34	X_{118}=20	X_{119}=25	X_{120}=25
X_{121}=30	X_{122}=11	X_{123}=11	X_{124}=20	X_{125}=19

DISCUSSION

After the first stage of the 60,000x24 stages sample the "best" solution had a total error of 168,993,744. However, after the 24th stage (always following the trail of better and better answers) the total error has decreased to 1,977,986 which is better but still not good.

However, after the second 60,000x24 solution attempt where the variable values that appear to be the same are pinned down the answer starts to improve. By stage 9 the total error is at 1266. Then by stage 11 the total error is 31. Stage 13 reports a total error of 0 and the correct answer which checks in all 125 equations.

Notice x_1=29 is in the solution. If it were pinned down incorrectly at 28 somewhere in the program, then the final solution would be wrong. Therefore, it's important not to pin down a variable value until the program is almost certain that that value must be in the true optimal solution

Would it be better to run this program longer and use a mode of 3 before pining any variable value down? Possibly modes of 3 or 4 might be better in some cases. (Conley 1993) and (Conley 1991) tried modes of 5 and 7 for averaging on the so called shortest route (or travelling salesman problems) problem connecting 150,200 and 600 points in a minimum distance. However, that did not involve solving systems of nonlinear equations and in the 1990's only samples of a few hundred or a thousand were practical so to mitigate the risks modes of 5 and 7 were selected.

However, today 60,000 samples can be drawn in an instant so there is inherently less risk and cost than there was in simulations last century. Would it be possible to use the sample means, or geometric means, or medians or harmonic mean or another measure of central tendency in these simulations? Also, instead of using .015 as a measure of deviation (which is a function of the range) would standard deviations work? Yes, all of these will work to close in on an optimal solution with statistical optimization (MSMCO) simulation.

Consider another system of nonlinear equations.

SECOND EXAMPLE

It is desired to find an all whole number solution to the following seven variable and eight equation nonlinear system in the range $1 \le x_i \le 720$ for i=1, 2, 3, 4, 5, 6, 7 and all whole numbers.

$$x_1^3 + x_1x_2 + x_3x_4 = 7906849 \quad (1)$$
$$x_2^3 + x_2x_4 + x_5x_7 = 136177802 \quad (2)$$
$$x_3^3 + x_1x_3 + x_1x_7 = 17362829 \quad (3)$$
$$x_4^3 + x_1x_4 + x_5x_6 = 239640510 \quad (4)$$
$$x_5^3 + x_1x_5 + x_6x_3 = 801750 \quad (5)$$
$$x_6^3 + x_1x_4 + x_5x_7 = 63705400 \quad (6)$$
$$x_7^3 + x_1x_4 + x_6x_3 = 347654206 \quad (7)$$
$$x1 + x_2^2 + x_4^2 + x_5x_7 = 711899 \quad (8)$$

Here the attempt is made to minimize

$$f(x_1,x_2,x_3,x_4,x_5,x_6,x_7) = \sum_{j=1}^{8} |L_j - R_j| \text{ subject to } 1 \le x_i \le 720$$

with Lj and Rj being the left hand side and the right hand sides of equations respectively for j=1,2 . . . 8. The fact that the number of equations exceeds the number of variables makes it probable (although not certain) that there is no solution. However, sometimes business people and/or scientists want (and can use) approximate solutions if an important problem has no exact solution.

Therefore, it is decided to use regular multi stage Monte Carlo optimization without modal averaging. Also, this feasible solution space has about 10^{20} values which is smaller compared to the previous problem of approximately 10^{212} feasible values.

Therefore a 40 stage MSMCO simulation drawing 1000000 sample answers at each stage is tried. The best answer produced was

x_1=197 x_2=514 x_3=258 x_4=621 x_5=88 x_6=399 x_7=703

which has an error of 1 in equation 8 and no errors in the other seven equations. Therefore, it appears that there is not an exact solution. However, in a practical application this small error might be acceptable.

Note that extended precision calculation was employed somewhat because of the large number of figures in some of the constants. Figure 2 gives a partial geometric and statistical view in two dimensions of MSMCO trying to close in on a useful approximate solution with very little error.

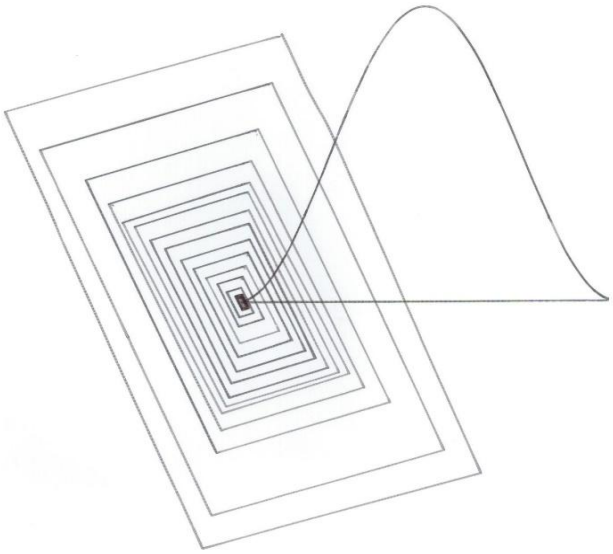


Figure 2. A Multi Stage Monte Carlo Optimization Simulation Looking for the Minimum Error Solution

CONCLUSION

Presented here were two systems of nonlinear equations. Multi stage Monte Carlo optimization (MSMCO) did not solve the seven variable eight equation system for an exact solution, although it did come up with an approximate solution. The difficulty appears to be that there is no exact solution to be found. This can happen in big business, small business, economics and science where there are so many goals in relation to the number of variables to be met (and customers to be satisfied). However, MSMCO simulation can be useful providing approximate solutions when no true optimal is available. Also, more equations (8) than variables (7) usually means no exact solution.

The featured example of 125 nonlinear equations with 125 variables did have an exact solution and MSMCO (or statistical optimization) did find it in four minutes of computer run time making heavy use of statistical averaging. Does this mean all very complex 125 variable nonlinear problems are solvable? Probably not yet, or ever, perhaps. However, once it becomes more widely known that local optimals (which are the real worry) tend to pile up around the true optimal and/or lead a path to the optimal, statistical optimization can be applied to so many fields (Conley 2013).

The statistical averaging of MSMCO simulations is crucial to solve the very large optimization problems if there are few solutions. The authors (Anderson 2003), (Anderson, Sweeney 1999), (Black 2014), (Conley 2013), (Conley 2016a), (Conley 2016b), (Conley 2016c), (Hayter 2002), and (Keller, Warrack 2003) present much useful statistical information for researchers interested in this statistical approach to simulation for large optimization problems.

One example would be to do 100 MSMCO solution tries. If they all fail to find the solution, just average them all (coordinate-wise) and keep going. In a sense that was what was done here on the 125x125 problem. Hundreds of solution attempts failed. However, the average of them led the way to the true optimal by the end of the simulation. There are patterns in the sampling distributions of optimization problems and statistical optimization will find them and through additional averaging close in on the optimal.

The 75 equation with 300 variable nonlinear system of equations that was solved in (Conley 2016) was a challenging problem. However, because the number of variables greatly exceeded the number of equations, there were "solutions" in many locations of the 300 dimensional space. Therefore, locating one of these was just a matter of time (or minutes) with multi stage Monte Carlo optimization (MSMCO) simulation. However, the 125 variable with 125 equation nonlinear system presented here has probably only one whole number solution, so for the MSMCO simulation one has to be very careful navigating through 125 dimensional space. This is where the statistical averaging becomes so important. One could even argue that our solution algorithm tried twelve major MSMCO runs and they all produced approximate "wrong answers." However, the continual successive averaging of them (successive approximation) was the exact solution with no error in any of the 125 equations. Even though the true optimal was found, computer statistical processes do have risks. However, these risks can be managed and mitigated in science, business and engineering. Additionally, Figures 3, 4, and 5 show three dimensional representations of MSMCO maximizing and minimizing functions.

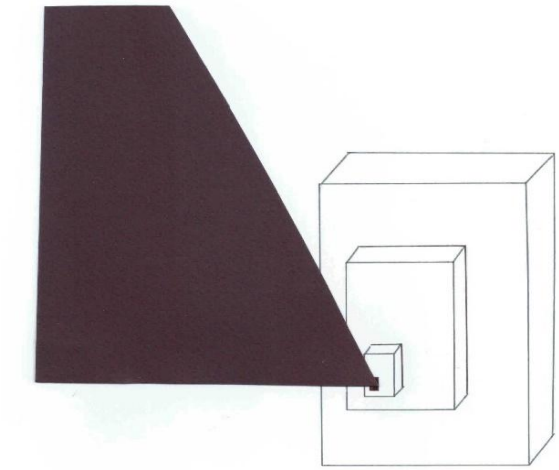


Figure 3: Optimizing with Multi Stage Monte Carlo

Almost any computer language can be used to write the MSMCO simulation optimization programs. The author has used six different programming languages on these programs over the years.

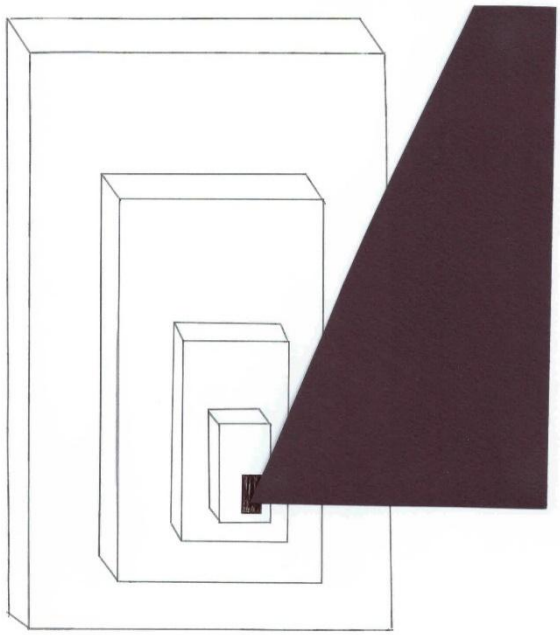


Figure 4: Three-Dimensional Search with MSMCO

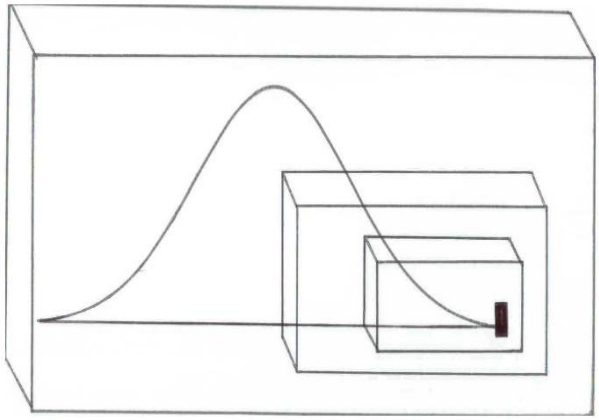


Figure 5: Closing in on the Solution with Statistical Optimization

REFERENCES

Anderson, T. W. 2003. *Multivariate Statistical Analysis*. 3rd edition. Wiley and Sons, New York.

Anderson, D. R., Sweeney, D. J., Williams, T. A. 1999. *Statistics for Business and Economics*. 7th edition. Southwestern College Publishing, Cincinnati, Ohio.

Black, K. 2014. *Business Statistics for Contemporary Decision Making*. 8th edition. John Wiley & Sons, New York.

Conley, W. C. 1991. "Programming an Automated Punch or Drill." *International Journal of Systems Sciences*, Vol. 22, No. 11, pp. 2039-2056.

Conley, W. C. 1993. "Multi Stage Monte Carlo Optimization Applied to a Six Hundred Point Traveling Salesman Problem." *International Journal of Systems Science*, Vol. 24, No. 6, pp. 609-626.

Conley, W. C. 2013. "Transportation and Minimum Content Problems" *Proceedings of European Simulation and Modelling Conference*. ESM, October 23-25, Lancaster, England. EUROSIS, Ostend, Belgium, pp. 325-330.

Conley, W. C. 2016a. "Eleven Dimensional Correlation Analyses." *Proceedings of the 2016 European Simulation and Modelling Conference*. ESM2016, Las Palmas, Spain, EUROSIS, Ostend, Belgium, pp. 10-14.

Conley, W. C. 2016b. "Finding Information in the Midst of Chaos." *Proceedings of the 2016 European Simulation and Modelling Conference*, ESM2016, Las Palmas, Spain, EUROSIS, Ostend, Belgium, pp. 23-27.

Conley, W. C. 2016c. "A 75x300 System Solved with Multi Stage Monte Carlo Algebra." *Proceedings of the 2016 European Simulation and Modelling Conference*, ESM2016, Las Palmas, Spain, EUROSIS, Ostend, Belgium, pp. 71-74

Hayter, A. J. 2002. *Probability and Statistics for Engineers and Scientists*. 2nd edition. Duxbury Press, Pacific Grove, CA.

Keller, G. and Warrack, B. 2003. *Statistics for Management and Economics*. 6th edition. Thompson Brooks/Cole, Pacific Grove, CA.

BIOGRAPHY

WILLIAM CONLEY received a B.A. in mathematics (with honors) from Albion College in 1970, an M.A. in mathematics from Western Michigan University in 1971, an M.Sc. in statistics in 1973 and a Ph.D. in mathematics-computer statistics from the University of Windsor in 1976. He has taught mathematics, statistics, and computer programming in universities for over 30 years. He is currently a professor emeritus of Business Administration and Statistics at the University of Wisconsin at Green Bay. The developer of multi stage Monte Carlo optimization and the CTSP multivariate correlation statistics, he is the author of five books and more than 230 publications world-wide. He is a member of the American Chemical Society, a fellow in the Institution of Electronic and Telecommunication Engineers and a senior member of the Society for Computer Simulation. He was also elected to Omicron Delta Epsilon (ODE) the national economics honorary. He gave a speech and presentation on his research at the 7th International Conference on Genetic Algorithms in July 1997 in the United States.

STATISTICAL OPTIMIZATION APPLIED TO A LARGE NONLINEAR SYSTEM OF EQUATIONS

William Conley
Austin Cofrin School of Business
University of Wisconsin at Green Bay
480B Wood Hall
Green Bay, Wisconsin 54311
U.S.A.
E-mail: Conleyw@uwgb.edu

KEYWORDS
Complex systems, nonlinear equations, averaging of approximations, exact solutions, error reduction.

ABSTRACT

Linear algebra, linear programming and even least squares forecasting which produces a system of linear normal equations that can be solved in forecasting problems, are all useful in many settings. However, so much of the real world of big business, science, engineering, economics and correlation analysis deals with factors and variables and systems of equations that should be nonlinear.

However, in the early part of the twentieth century, computers were not available so linear theories dominated because nonlinear systems were harder to solve with classical mathematics.

Today, in the early part of our twenty-first century, computers are powerful enough and so ubiquitous that virtually any scientist, business person or economist has enough computing power on her/his desk (or smart phone) to tackle difficult nonlinear systems in applications. Presented here is a 200 variable 200 equation nonlinear system of equations which is worked on and solved using the statistical optimization simulation approach.

INTRODUCTION

Consider the system of equations:

$x_1 + 2x_2 + x_3 + x_4 = 32$
$3x_1 + x_2 + x_3 + 2x_4 = 53$
$x_1 + x_2 + 10x_3 + x_4 = 54$
$x_1 + x_2 + x_3 + x_4 = 27$

After taking an algebra and /or a linear algebra course one could solve this system of linear equations using substitution or cancellation. However, a third approach would be to transform this four by four system to minimize: $f(x_1,x_2,x_3,x_4) = |x_1 + 2x_2 + x_3 + x_4 - 32| + |3x_1 + x_2 + x_3 + 2x_4 - 53| + |x_1 + x_2 + 10x_3 + x_4 - 54| + |x_1 + x_2 + x_3 + x_4 - 27|$ and write a statistical optimization (multi stage Monte Carlo optimization) computer simulation to rocket across the sampling distribution of $f(x_1,x_2,x_3,x_4)$'s feasible solution space to the solution $x_1=7$, $x_2=5$, $x_3=3$ and $x_4=12$. Let us try a much larger and nonlinear system with our computer simulation Monte Carlo optimization approach and continual "averaging" of the approximate preliminary solutions in the early stages to lead us to a solution if one exists.

A 200 VARIABLE 200 NONLINEAR EQUATION SYSTEM

An all whole number solution is required for the solution to the nonlinear system of equations

$x_i^3 + x_{201 - i} \cdot x_{200 - i} + x_{199 - i} + x_{198} - i = C_i$

for i = 1, 2, 3, . . . 197

and

$x_{198}^3 + x_{71}\, x_{67}\, x_{198} = C_{198}$

$x_{199}^3 + x_{57}\, x_{112}\, x_{199} = C_{199}$

$x_{200}^3 + x_{26}\, x_{96}\, x_{200} = C_{200}$

for $1 \leq x_i \leq 100$ for i= 1, 2, 3, . . . 200

and again, all X_i's must be whole numbers for the application at hand. There are 1×10^{400} feasible solutions in its sampling distribution.

All of the constants C_i's (the values on the right hand sides of the equations) are presented here as follows in the pattern:

C_1	C_2	C_3	C_4	C_5
C_6	C_7	C_8	C_9	C_{10}
"	"	"	"	"
"	"	"	"	"
"	"	"	"	"
"	"	"	"	"
"	"	"	"	"
C_{196}	C_{197}	C_{198}	C_{199}	C_{200}

Table 1: Right Hand Side of Equations Constants

664580	1002521	87144	307979	6570
191865	576865	805606	8217	8709
15353	3361	38311	80900	377986
102389	104188	512445	237045	266706
92038	1578	329271	8455	14023
284766	577534	12866	5337	49416
1108	4335	175689	705053	27145
615	912826	729665	706779	51227
111544	283104	196445	532348	362399
3273	92282	63230	2210	20303
239810	18224	3809	275839	277175
5895	14212	835744	554752	315979
683180	11110	139666	76788	1727
264076	59597	389225	706951	705680
79735	614493	16569	5536	47062
45438	1006010	6448	617624	308552
208196	316151	595887	1004639	364215
17044	36347	456701	274802	3083
2854	105282	7724	1003957	82464
223830	597905	374789	730082	318
79933	18791	280775	498221	943830
596969	92978	55735	43233	389225
12992	5090	16461	5092	660462
380500	258574	116430	40101	555420
836393	47980	942660	219442	1001189
439358	262472	3985	10692	4501
622158	18767	3040	41846	175965
24626	731248	457607	302497	18327
232670	165205	832449	456929	60307
231244	6764	593118	28588	99107
1759	390017	28339	51210	288427
306637	6539	1003855	595950	2955
62799	93299	11512	1020	1114
5486	5130	1937	10765	85385
238467	614293	177364	1005438	1001584
5451	4809	80237	7389	94146
1004031	1004973	3905	168652	732388
146743	15272	804593	36109	804814
176077	12555	132782	1004804	287949
831130	425010	54072	946778	464881

Therefore, letting L_i and C_i be the left and right hand side of equation I respectively, the 200 x 200 nonlinear system of equations is transformed to minimize

$f(x_1\ x_2\ x_3\ \ldots\ x_{200}) = \sum_{i=1}^{200} |L_i - C_i|$ subject to $1 \le x_i \le 100$ and all variables must be whole numbers. The plan is to minimize it all ways down to zero solving the system exactly if possible!

THE SOLUTION APPROACH OR ALGORITHM

In the world of theoretical mathematics an algorithm is a theorem that is instructive, meaning it proves the solution to a problem exists by finding it. The simplex algorithm in linear programming is an example. However, the computer field has begun generalizing the definition of algorithm to mean solving problems by finding a solution with a computer program.

The algorithm used here was to do two multi stage Monte Carlo optimization (MSMCO) simulation tries of 35 stages each in the x_is equal 1 to 100 region for i= 1, 2, 3, . . . 200 drawing 100,000 feasible solutions at each stage which is ever decreasing in size and free floating to pursue the optimal solution region. Real value arithmetic is used here. However, whole number solutions are required. Therefore, after each pair of 35 stage runs is completed they are compared coordinate-wise and if any two x_i's are less than .015 apart they are averaged and then rounded to the nearest whole number and pinned down at the value for the rest of the simulation. The aforementioned comparison of two is done eleven more times and more and more x_i values are pinned down. For example, if the two x_{200} values were 60.999 and 61.002 then these would be "modal averaged" to be fixed at x_{200} = 61 for the rest of the program. After the first sample of 100,000 solution tries with this approach the total error was about 49,000,000. However, after the two stages of solution tries (numbers 1 and 2 of the 25 tries) a few of the variable values were pinned down (reducing the search region). Then after the next two tries a few more were pinned down, etc.

By the end of this simulation program (which took 30 minutes of run time on the author's desk top office computer) that drew 25x35x100,000 = 87,500,000 sample answers the total error of all the equations was down to 0.00 with the solution that follows here in the pattern.

x_1	x_2	x_3	x_4	x_5
x_6	x_7	x_8	x_9	x_{10}
"	"	"	"	"
"	"	"	"	"
"	"	"	"	"
"	"	"	"	"
"	"	"	"	"
x_{196}	x_{197}	x_{198}	x_{199}	x_{200}

This last pass through the feasible solution space (after 12 sets of two with modal averaging comparison) was not compared with any other simulation and its answer was the exact solution given here, which checks in all 200 equations.

The 25 stage errors are presented are given here also.

Table 2: Chart of Selected Stage Errors

At End of Stage	Total Stage Error
1	3638230.00
2	4041514.00
3	1776752.50
4	2285310.00
5	1047484.62
6	1255802.00
7	711014.50
8	663726.31
9	192895.43
10	295706.81
11	93997.07
12	146752.03
13	44602.91
14	36257.96
15	20141.77
16	26055.845
17	13090.47
18	17737.10
19	4370.25
20	19495.58
21	5810.13
22	5075.04
23	200.35
24	8.57
25	0.00

Table 3: The Solution

87	100	44	67	6
57	83	93	19	15
23	6	33	43	72
46	47	80	61	64
45	9	69	20	23
65	83	20	1	36
5	16	56	89	30
8	97	90	89	37
48	65	58	81	71
9	45	86	11	27
62	25	13	65	65
14	19	94	82	68
88	18	51	42	4
64	39	73	89	89
43	85	25	8	34
34	100	14	85	67
59	68	84	100	71
25	33	77	65	11
6	47	18	100	42
60	84	72	90	3
43	23	65	79	98
84	45	38	35	73
23	3	24	16	87
72	63	48	33	82
94	36	98	60	100
76	64	12	19	8
84	23	5	34	56
29	90	77	67	23
61	54	94	77	39
61	18	84	30	46
11	73	29	37	66
67	12	100	84	10
39	44	14	5	9
14	5	10	22	44
62	85	56	100	100
17	15	43	19	45
100	100	3	55	90
52	24	93	33	93
56	22	50	100	66
94	75	24	98	61

COMMENTS

One could argue that the first 87,000,000 feasible solutions looked at failed to find the answer. However, the "modal averaging" of all those approximate solutions put the simulation very near the "true optimal (no error)" and the last 500,000 sample answers closed in and found it.

There appear to be two "risks" in this approach. They are local optimals and pinning down an x_i with the wrong value early on in the simulation. These will lead to the wrong answer (although it still could be a useful approximation).

The author has studied local optimals in optimization problems for about 50 years and they appear to pile up around the true optimal. That can be exploited to increase the odds of finding the right answer as was done here.

Also, it is crucial to not pin down an x_i incorrectly at any point in the repeated multi stage Monte Carlo optimization (MSMCO) simulations. This program was lucky to not make that mistake. However, enough samples were drawn to be pretty sure it would avoid that problem.

Would it be better to use a mode of 3 or 4 or 5 or 6 or 7 to be sure? (Conley 1991) used modes of up to 7 last century on difficult transportation problems to minimize one distance equation because computers were so slow then that small samples had to be drawn. However, our 21st century provides powerful computers. Also, the deviation measure for determining that two x_i values were close enough to modal average them and pin them down was set at .015 for the bound. That deviation value could be changed if the simulation is having trouble converging to the solution (or a good approximation). Also, instead of a fixed bound like .015 for all of the 200 x_i variables, if the mode is set at three through seven, some function of the sample ranges or mean absolute deviations or standard deviations coordinate-wise could be used. Additionally, in place of the mode for an average, one could use the sample mean, median, geometric mean, harmonic mean or a weighted mean coordinate-wise. Any and all of these measures of average and deviation should work to help organize the closing in to the optimal solution region.

(Conley 1994) experimented with sample means, weighted means and modes last century along with using fixed ranges (as was done here) and sample standard deviations (again last century).

However, in our 21st century there is so much inexpensive computing power that any and all of the measures of average and deviation should work in combination to close in effectively on the true optimal or a useful approximation.

Therefore, can all nonlinear multivariate optimization problems be solved with statistical optimization? No, probably not (so far anyway). However, on difficult optimization problems if nothing else works, it might be worth a try.

The linear algebra and linear programming subjects have powerful theorems ("theoretical mathematics") to guarantee and find solutions very effectively. These subjects are of obvious value. However, their major "risk" is the all too frequent approach of linearizing essentially nonlinear optimization problems, to guarantee an exact solution to perhaps the "wrong" question?

However, the "good news" in this discussion is that now we have more viable solution techniques (linear and nonlinear too) to chose from when complex quantitative problems arise in many different fields.

CONCLUSION

It is "risky" to use computer simulation but risk can be managed successfully in the insurance fields with careful and accurate probability and statistics calculation by the actuaries and risk can be managed when using simulation to solve optimization problems also.

Presented here was a 200 variable 200 equation nonlinear system of equations that required an all whole numbers solution and had a feasible solution space of $1x10^{400}$ answers. It found the right one (exact solution in 30 minutes of computer time (drawing and testing 87,500,000 of the feasible solutions and keeping track of and averaging them).

Last century this simulation would have taken weeks or months or years of run time on a desk top computer. There exists computers in our 21st century that would run this simulation optimization program in a tiny fraction of a second instead of 30 minutes. The new IBM super computer (Smith and Loehrke 2018) does $2x10^{17}$ calculations per second. This means it could run this simulation rocketing across the feasible solution space to the true optimal in about .00000001 of a second instead of the 30 minutes reported here.

Machine learning and artificial intelligence can benefit from computer simulation. Further reading on statistics and some other successes with optimization (Verma, et al 2018) or multi stage Monte Carlo optimization (MSMCO) can be found in (Anderson 2003), (Black 2014), (Conley 1991), (Conley 2013), (Keller and Warrack 2003), (Anderson, Sweeney, Williams, 1999), (Conley 1993), (Conley 1994), (Conley 2016a, (Conley 2016b, (Conley 2016c), and (Hayter 2002).

Even though this statistical optimization (MSMCO) found the exact solution to the large optimization problem presented here, one of its better uses for big business, economics or engineers and scientists may be its ability to

find a useful approximate solution to hopelessly complex multivariate problems. This can include, of course, problems that are important to practitioners where it is known that there is no solution, so the goal at the outset is to find a useful approximate solution in a cost effective manner.

Figures 1 and 2 below further illustrate MSMCO with two and three variables.

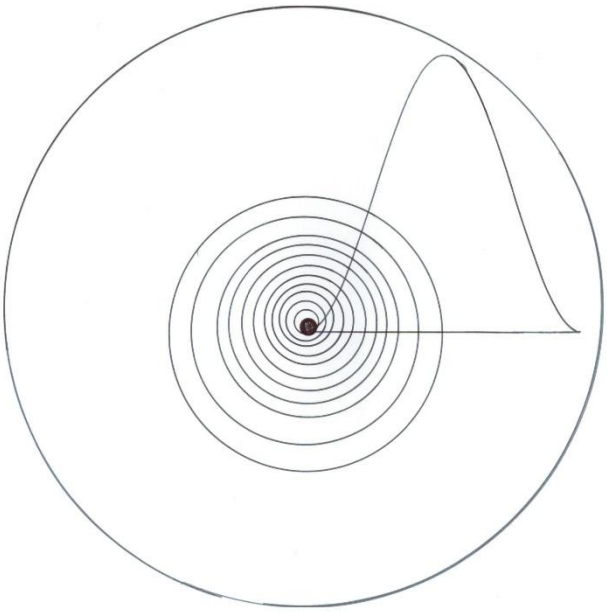


Figure 1: Minimizing with Two Dimensions

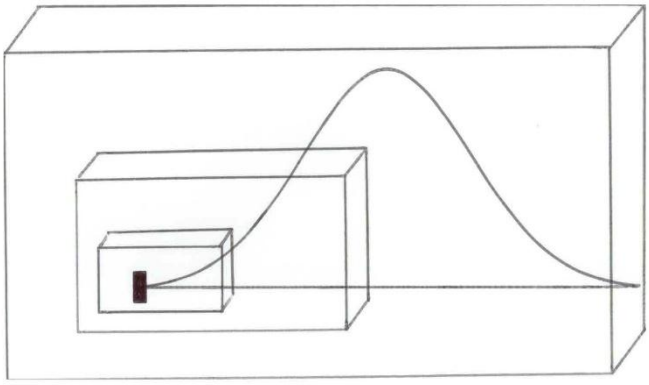


Figure 2: Minimizing a Three Variable Function

REFERENCES

Anderson, D. R., Sweeney, D. J., Williams, T. A. 1999. *Statistics for Business and Economics*. 7th Edition. Southwestern College Publishing, Cincinnati, Ohio.

Anderson, T. W. 2003. *Multivariate Statistical Analysis*. 3rd Edition. Wiley and Sons, New York.

Black, K. 2014. *Business Statistics for Contemporary Decision Making*. 8th Edition. John Wiley and Sons, New York.

Conley, W. C. 1991. "Programming an Automated Punch or Drill." *International Journal of Systems Sciences*, Vol. 22, No. 11, 2039-2056.

Conley, W. C. 1993. "Multi Stage Monte Carlo Optimization Applied to a Six Hundred Point Traveling Salesman Problem." *International Journal of Systems Science, Vol. 22, No. 11, 2039-2056.*

Conley, W. C. 1994. "A 50 Variable Nonlinear System." *Proceedings of the 1994 Summer Computer Simulation Conference*, SCS. San Diego, CA.

Conley, W. C. 2013. "Transportation and Minimum Content Problems." *Proceedings of European Simulation and Modelling Conference*. ESM, October 23-25, Lancaster, England. EUROSIS, Ostend, Belgium, 325-330.

Conley, W. C. 2016a. "Eleven Dimensional Correlation Analyses." *Proceedings of the 2016 European Simulation and Modelling Conference.* ESM 2016, Las Palmas, Spain, EUROSIS, Ostend, Belgium, 10-14.

Conley, W. C. 2016b. "Finding Information in the Midst of Chaos." *Proceedings of the 2016 European Simulation and Modelling Conference*, ESM2016, Las Palmas, Spain, EUROSIS, Ostend, Belgium, 23-27.

Conley, W. C. 2016c. "A 75X300 System Solved with Multi Stage Monte Carlo Algebra." *Proceedings of the 2016 European Simulation and Modelling Conference*, ESM2016, Las Palmas, Spain, EUROSIS, Ostend, Belgium, 71-74.

Hayter, A. J. 2002. *Probability and Statistics for Engineers and Scientists*. 2nd Edition. Duxbury Press, Pacific Grove, CA.

Keller, G. and Warrack, B. 2003. *Statistics for Management and Economics*. 6th Edition. Thompson Brooks/Cole, Pacific Grove, CA.

Smith, M. and Loehrke, J. 2018. "200000 Trillion Calculations per Second." USA Today, June 22, 2018, Weekend Edition, Arlington, VA, p. 1.

Verma, A., Jaso, S., Lopez, J., Chen, J. and Bracht, M. 2018. "Multi-Scale Simulation and Optimization: Natural Gas to Products." *Proceedings of the 16th International Industrial Simulation Conference.* ISC 2018, Ponta Delgada, Portugal, EUROSIS, Ostend, Belgium, 13-15.

BIOGRAPHY

WILLIAM CONLEY received a B.A. in mathematics (with honors) from Albion College in 1970, an M.A. in mathematics from Western Michigan University in 1971, a M.Sc. in statistics in 1973 and a Ph.D. in mathematics-computer statistics from the University of Windsor in 1976. He has taught mathematics, statistics, and computer programming in universities for over 30 years. He is currently a professor emeritus of Business

Administration and Statistics at the University of Wisconsin at Green Bay. The developer of multi stage Monte Carlo optimization and the CTSP multivariate correlation statistics, he is the author five books and more than 230 publications world-wide. He is a member of the American Chemical Society, a fellow in the Institution of Electronic and Telecommunication Engineers and a senior member of the Society for Computer Simulation. He was elected to Omicron Delta Epsilon (ODE) the national economics honorary and Kappa Mu Epsilon (KME) the national mathematics honorary. He gave a speech and presentation on his research at the 7th International Conference on Genetic Algorithms in July 1997 in the United States.

ON PARALLEL CONFORMANT PLANNING AS AN OPTIMIZATION PROBLEM

Adam Galuszka, Tomasz Grzejszczak, Jaroslaw Smieja
and Adrian Olczyk
Institute of Automatic Control
Silesian University of Technology
44-100, Gliwice, Poland
E-mail: {adam.galuszka, tomasz.grzejszczak,
jaroslaw.smieja, adrian.olczyk}@polsl.pl

Jerzy Kocerka
Automatix sp. z o.o.,
Katowice,
Poland
E-mail: jerzy.kocerka@polsl.pl

KEYWORDS
Conformant Planning, Parallel Planning, Linear Programming Computational Complexity , Operations Research , Decision support systems.

ABSTRACT

Software testing and the quality of the requirements plays the the important role in quality assurance and risk management problems. In this paper we propose to model a class of a decision process related to these problems as an artificial intelligence conformant planning problem. Planning in Artificial Intelligence is a problem of finding a sequence of actions that transforms a given initial state of the problem to a desired goal situation. Conformant planning is a problem of searching for a non-conditional plan for a problem with an uncertain initial state. A parallel plan is a plan in which some actions can be executed in parallel, usually leading to a decrease of the plan execution time. We consider a problem of finding a parallel conformant plan which is computationally difficult. To avoid this difficulty, a heuristic of transformation to Linear Programming Problem, illustrated by an example, is proposed.

INTRODUCTION

One of the important part of the quality assurance and risk management is software testing and the quality of the requirements. Due to the increasing complexity of systems number of functions (and requirements) that need to be verified in the testing process can reach up to several thousand. To meet the expected delivery schedules it is important to make development and testing process as efficient as possible, but even then testing is one of the most time-consuming parts of the development process which takes 40-70% of the overall effort (Kosindrdec and Daengdej 2005). An important part of this effort is test case development which requires highly skilled engineers knowing testing process, test environment and the domain to be able to analyze and understand the requirements for the system under test.

Requirements engineering is a result of the iterative and cooperative process of analyzing problem (Loucopoulos and Karakostas 1995). Multiple actors are involved in this process including system engineers, software engineers but also nontechnical people.

Using natural language for requirements allows to focus on communication and knowledge instead of formal syntax or modelling. It allows also to:

- interact with other, nontechnical actors;
- use common tooling and processes for different kinds of requirements (functional, nonfunctional, organizational etc.);
- provide the first project artifacts very quickly (minimal learning curve);
- use common tools for change tracking and management;

For this reason most of the requirements (79%) are written in the common natural language, such as English, with only 21% using some kind of formalism (structured natural language like forms or templates or formalized language) (Mich et al. 2004). Despite many advantages, writing requirements in common language generates many challenges. The requirement should be precise, unambiguous and complete (Requirements Working Group 2015). Those characteristics are not always ensured when writing in natural languages. Due to this, preparing a test cases to verify if the tested software has been properly developed according to requirements, requires high skills, deep analysis and discussions between system engineers, software engineers and test engineer. Ambiguous or not precise requirements may lead also to problems with qualification of software/system behavior as a potential bug. In this work we propose to model a class of a decision process related to these problems above as an artificial intelligence planning problem.

Classical planning as Artificial Intelligence problem is formulated as a searching process leading to a sequence of agent's actions (called a *plan*) that transforms an initial agent environment (called *initial state of a planning problem*) to a desired goal situation (e.g. Bylander 1994, Rintanen 1999, Skrzypczyk 2010).

The problem becomes more complicated, if information about the modeled world is not sufficient to determine all facts necessary to describe an initial state of the world. Then, we say that the initial state of the problem is uncertain but can be represented by a set of possible initial states. A plan for solving such problem may take the form of actions that are executed conditionally, based on new information emerging during the search for the plan. The inflow of new information is modeled by the so-called sensory actions, in such way that the uncertainty of the information available is reduced by using information from the sensors. This

approach is called *conditional planning* (Smith and Weld 1998, Weld et al. 1998).

In some cases, information from sensors may be unavailable e.g. sensors are damaged or broken down, receiving sensory information is too expensive or dangerous. Then, it is reasonable to search for a plan that is a solution to the planning problem independently of possible initial states. This approach is called *conformant planning* (also Smith and Weld 1998, Weld et al. 1998). Both conditional and conformant planning are more difficult to solve than a classical planning (Bonet 2010).

The cases, in which more than one action can be applied in one planning step, i.e. some actions can be performed simultaneously, constitute a large class of important planning problems. Such problem formulation allows to model multi-agent and multi-robot environments and is called *parallel planning* (e.g. Ghooshchi et al. 2015). Combining *conformant* and *parallel planning* leads to a problem, in which many agents interact in an uncertain environment with no possibility of performing sensing actions. Finding a solution to a parallel conformant planning problem is more difficult than for previous problems. To avoid this difficulty, in the paper we propose a heuristics for transformation of the problem to a Linear Programming Problem, illustrated by an example.

Motivation

With respect to computational complexity terminology, planning with complete information is PSPACE-complete problem and planning in the presence of incompleteness is much more complicated. A high level of computational complexity of planning is the reason that different heuristic techniques are very popular in implementations of planning solvers (e.g. Palacios and Haffner 2009, Rosa et al. 2011, Galuszka and Swierniak 2010). One of the heuristics is a transformation of planning to a Linear Programming (LP) problem. The idea of representing planning problems by linear constraints and an objective function is not new in the literature (see e.g. Nareyek et al. 2005). In these cases, the planning problem takes the form of binary integer linear program. It implies that the only allowed values of variables are '0' and '1' and they correspond to false/truth values of planning problem predicates and actions. The computational efficiency of the approach is low (because of complexity of integer programming algorithms) and the solution can be found only for small size planning problems. Another approach, proposed by Bylander (1997) and developed in (Galuszka 2011), is to introduce additional linear constraints to LP problem. It allows to solve optimally some class of difficult planning problems using classical LP polynomial algorithms. The drawback of such approach is that not always the LP solution can be directly interpreted as a plan (what is followed by assumption P ≠ NP). Moreover, the size of LP problem increases very fast (but polynomially) with the number of planning problem variables.

Contribution

In the paper a heuristic of polynomial transformation of planning problem to LP problem, proposed in (Bylander 1997) and developed in (Galuszka 2011 and 2013), is extended to handle parallel conformant planning. The problem of transforming conformant planning problems with single action in each planning step to LP has been studied earlier (Galuszka et al. 2015). In chapter 2 planning and conformant planning problems are formulated, in chapter 3 the transformation of conformant planning to LP as well as an example are presented, in chapter 4 computational complexity of proposed approach is presented.

PRELIMINARIES

Following Bylander (1994) it is assumed that classical planning problem is denoted by P (also called STRIPS planning) and consists of four sets P={*C, O, I, G*}, where:

- *C* is a finite set of *conditions*,
- *O* is a finite set of actions, where each action $o \in O$ takes the form $c^+, c^- \rightarrow c_+, c_-$, where:
  - $c^+ \subseteq C$ are called *positive preconditions*,
  - $c^- \subseteq C$ are called *negative preconditions*,
  - $c_+ \subseteq C$ are called *positive postconditions*,
  - $c_- \subseteq C$ are called *negative postconditions*,
- $I \subseteq C$ is an *initial state*,
- $G = \{G_+, G_-\}$ is a *goal situation*, where $G_+ \subseteq C$ are positive conditions (i.e. are true) and $G_- \subseteq C$ are negative conditions (i.e. are false).

In order to include information that some conditions are unknown (assume *k* conditions can be true or false) in the description of the current problem state, one can introduce so called *k-states* proposed by Baral et al. (2000). In simple terms, the *k-state* is a pair (S, Σ), where S is the current problem state (so called *belief state*), and Σ is a set that consists of all possible initial states I (set of *belief states*). The number of possible initial states (*worlds*) is the size of Σ set, it depends on *k* and will be denoted by *w*:

$$w = |\Sigma| \leq 2^k. \quad (1)$$

For unknown initial state the set Σ consists of all states *s*, for which:

- condition $c \in C$ is true in the initial state (i.e. $c \in I$),
- condition $c \in C$ is false (i.e. $\neg c \in I$),
- if it is unknown whether condition $c \in C$ is true or false in the initial state, then the set Σ includes both states, for which this condition is true and false, respectively.

The initial state I can be potentially any state from states included in set Σ. It follows that planning problem with incomplete information about initial state (STRIPSIN) takes the form:

$$\Pi = \{C, O, \Sigma, G\}. \quad (1a)$$

The result of applying action to the current state is presented below, is based on (Baral 2000) and is adopted to STRIPSIN problem. For action *o*, *k-state* is described by a set $\{Result(S,<o>), Result((S, \Sigma),<o>)\}$, where $Result(S,<o>)$ is the same like in case with complete information, e.g.:

$Result(S, \{\}) = S$,

$Result(S, \{o\}) = (S \cup c_+) \setminus c_-$ if $c^+ \subseteq S \wedge c^- \cap S = \emptyset$;
S in opposite case,

$Result(S, <o_1, o_2, ..., o_n>) =$
$= Result(Result(S, <o_1>), <o_2, ..., o_n>),$
and for k-state:
$Result((S, \Sigma), <o>) = \{ Result(S, <o>) \mid S \in \Sigma \}.$

The plan $\Delta_C = <o_1, o_2, ..., o_n>$ solves conformant planning problem, if $Result((S, \Sigma), \Delta_C) = G$. Since all actions in Δ_C are ordered, Δ_C is called a *total order conformant plan*.

The *partial-order conformant plan* is denoted as $\Delta_{POC} = \{\Delta_{SetC}, \pi\}$, where $\Delta_{SetC} = \{o_1, o_2, ..., o_n\}$ is the set of actions, and π is the non-returnable partial order defined on Δ_{SetC} (compare Backstrom 1998). So, a *partial-order conformant plan* is a compact representation of a set of possible *total* ordered plans.

The *parallel partial-order conformant plan* is denoted as $\Delta_{PPOC} = \{\Delta_{SetC}, \pi, \#\}$, where $\{\Delta_{SetC}, \pi\}$ is Δ_{POC}, while # is a symmetrical relation, defined on the set Δ_{SetC}. $\# \subseteq (\pi \cup \pi^{-1})$ is called a non-concurrency relation and it indicates which actions cannot be applied in parallel.

Example. To illustrate a parallel conformant planning problem consider the following simple *bomb in the toilet* problem (Son Tran Cao et al. 2005) with one action containing conditional effects:

Dunk(P): preconditions: package(P), bomb(B)
effects: if in(P, B) then defused(B), (2)

meaning that if there is a bomb B and there is a package P then action *dunk* causes that bomb B is defused if it was in package P. So if there is no bomb in package, the action *dunk* has no effects.

Action model in formula (2) is different than classical STRIPS action. It is caused by a fact that actions effects are formulated conditionally with general schema: actions causes *set1_of_conditions* if *set2_of_conditions.* Please note that an action defined in such way has no preconditions but action effects are formulated conditionally. It implies that preconditions are indirectly defined by effects, so action *Dunk* is equivalent to two classical STRIPS actions:

$Dunk_1$(P): pre: package(P), bomb(B), in(P, B);
eff: defused(B)
$Dunk_2$(P): pre: package(P), bomb(B), not(in(P, B));
eff: no effects

Now, let us consider the following problem Π_{BT} with two possible initial states (bomb is in package 1 or in package 2):

$\Pi_{BT} = \{C_{BT}, O_{BT}, \Sigma_{BT}, G_{BT}\}$, (3)

where:
C_{BT} = *package(P1),package(P2),bomb(B),in(P1,B),in(P2,B), defused(B)*},
O_{BT} = { *Dunk* },
Σ_{BT} = {{ *package(P1),package(P2),bomb(B),in(P1,B)*},
{*(package(P1),package(P2),bomb(B),in(P2,B)*}},
G_{BT} = { *defused(B)* }.

The conformant plan that solves the Π_{BT} problem is:

$\Delta_{CBT} = <$ Dunk(P1), Dunk(P2)$>$ or

$\Delta_{CBT} = <$ Dunk(P2), Dunk(P1)$>$. (4)

The partial-order conformant plan that solves the Π_{BT} problem is:

$\Delta_{POCBT} = \{$ Dunk(P2), Dunk(P1)$\}$. (5)

If actions in Δ_{POCBT} can be performed in parallel ($\#_{BT} = \emptyset$), then $\Delta_{POCBT} = \Delta_{PPOCBT}$ and the problem is solved in one step. *End of example*.

TRANSLATION TO LINEAR PROGRAMMING PROBLEM

Following (Bylander 1997), the transformation from planning to Linear Programming is based on mapping of conditions and operators in each plan step to variables. Truth values of conditions are mapped to "0" and "1" for the planning without incompleteness, and to any values between "0" and "1" for planning with incomplete information. The objective function reaches the maximum, if the goal situation is true in the last step of planning.

If l is the number of planning steps and w is the number of possible initial world states then variables of problem (3) for conditions are:

$c_1(i)$=package(P1,i), $c_2(i)$=package(P2,i),
$c_3(i)$=bomb(B,i), $c_4^w(i)$ = in(P1,B,i) ^{w},
$c_5^w(i) = in(P2,B,i)^w$, $c_6^w(i) = defused(B,i)^w$,
$i = 0, 1, \ldots l$, $w = 1, 2$, (6)

for actions:

$o_1^w(i) = Dunk_1(P1,i)^w$, $o_2^w(i) = Dunk_2(P1,i)^w$,
$o_3^w(i) = Dunk_1(P2,i)^w$, $o_4^w(i) = Dunk_2(P2,i)^w$,
$i = 0, 1, \ldots l-1$, $w = 1, 2$. (7)

The initial state is a disjunction of two possibilities. It is modelled by a set of equality constraints:

package(P1,0) = 1, package(P2,0) = 1, bomb(B,0) = 1,
in(P1,B,0) ^{1} = 1, in(P2,B,0) ^{1} = 0, defused(B,0) ^{1} = 0,
in(P1,B,0) ^{2} = 0, in(P2,B,0) ^{2} = 1, defused(B,0) ^{2} = 0, (8)

Goal state G_{BT} is reached if condition *(defused B)* is *true* in last planning step in each world, so the objective function of LP is:

Max $\leftarrow$ f = defused(B,l)^1 + defused(B, l) ^{2}. (9)

It leads to following formulation of optimization problem: *Find minimal number of planning steps l, such that f* = 2.

The set of constraints is given by:
- actions can be applied if preconditions are true (these are inequality constraints), so basing on formula (3) we have:

package(P1,i) $\geq Dunk_1(P1,i)^w + Dunk_2(P1,i)^w$,
package(P2,i) $\geq Dunk_1(P2,i)^w + Dunk_2(P2,i)^w$,
r * bomb(B,i) $\geq Dunk_1(P1,i)^w + Dunk_2(P1,i)^w$ +
$+ Dunk_1(P2,i)^w + Dunk_2(P2,i)^w$,
in(P1,B,i) $^w \geq Dunk_1(P1,i)^w$,
in(P2,B,i) $^w \geq Dunk_1(P2,i)^w$,
(1 - in(P1,B,i) ^{w}) $\geq Dunk_2(P1,i)^w$,
(1 - $in(P2,B,i)^w$) $\geq Dunk_2(P2,i)^w$,
$i = 0, 1, 2, \ldots l$-1,w=1,2 , (10)

r is a natural number indicating how many actions can be performed in parallel (in our example $r = 2$),
- changes of variables for conditions due to action application (these are equality constraints), so basing on formula (3) we have:

$$\text{defused}(B, i+1)^w = \text{defused}(B,i)^w + \\ + \text{Dunk}_1(P1,i)^w + \text{Dunk}_1(P2,i)^w$$
$$i = 0, 1, 2, \ldots l\text{-}1,\ w = 1,2. \qquad (11)$$

The equality constraint (11) should be studied more carefully. If actions $Dunk_1(P1,i)^w$ and $Dunk_1(P2,i)^w$ are applied in parallel in the same planning step, then the value of condition $defused(B, i+1)^w$ becomes infeasible. In this case, one should introduce an additional balancing variable for each condition in each planning step to avoid infeasibility:

$$\text{defused}(B, i+1)^w + \text{defused}(B, i+1)_b{}^w = \\ = \text{defused}(B,i)^w + \text{Dunk}_1(P1,i)^w + \text{Dunk}_1(P2,i)w,$$
$$i = 0, 1, 2, \ldots l\text{-}1,\ w = 1, 2. \qquad (12)$$

Basing on formulas (6) to (12) it is easy derive general formulas for any STRIPSIN problem (1). Finally, the LP problem for a given problem takes the following form:

$$\max_x \leftarrow f^T x \qquad Ax \le b \\ A_{eq}x = b_{eq} \\ 0 \le x \le 1 \qquad (13)$$

for which nonzero elements of matrices A , A_{eq} are shown in Figs 1 and 2.

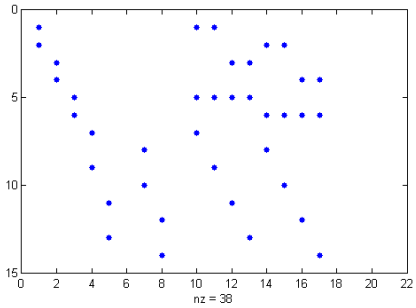


Fig. 1. Nonzero elements of Matrix A for (14).

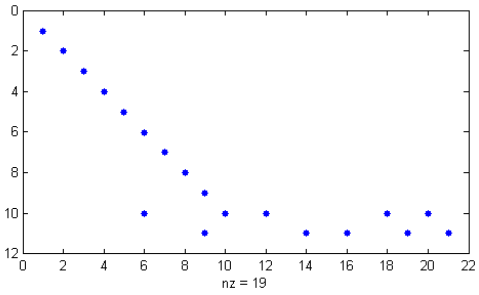


Fig. 2. Nonzero elements of Matrix Aeq for (14).

Table 1 presents the optimal solution of the problem (13) as well as two additional test problems with possible initial states given by set of equalities (14) and (15):

$$\text{in(P1,B,0)}^1 = 1,\ \text{in(P2,B,0)}^1 = 0,\ \text{defused(B,0)}^1 = 0, \\ \text{in(P1,B,0)}^2 = 1,\ \text{in(P2,B,0)}^2 = 1,\ \text{defused(B,0)}^2 = 0; \qquad (14)$$
$$\text{in(P1,B,0)}^1 = 1,\ \text{in(P2,B,0)}^1 = 0,\ \text{defused(B,0)}^1 = 0, \\ \text{in(P1,B,0)}^2 = 0,\ \text{in(P2,B,0)}^2 = 0,\ \text{defused(B,0)}^2 = 0. \qquad (15)$$

In the first one (14) there is a bomb in first package but it is uncertain whether it is in the second package, in second one (15) there is no bomb in second package but it is uncertain whether it is in first package.

It should be noted that values of variables for actions are binary integer, so the solution presented in table 1 can be directly interpreted as a plan:

$$\Delta_{PPOCBT} = \{ Dunk(P2),\ Dunk(P1)\}.$$

In the opposite case, one should apply additional heuristics or methods that lead to a binary integer solution.

Table 1. Optimal solution *xopt* for problems (14,15,16).

	LP variable	problem (13)	problem (14)	problem (15)
	package(P1,0)	1	1	1
	package(p2,0)	1	1	1
	bomb(B,0)	1	1	1
world 1	in(P1,B,0)	1	1	1
	in(P2,B,0)	0	0	0
	defused(B,0)	0	0	0
world 2	in(P1,B,0)	0	1	0
	in(P2,B,0)	1	1	0
	defused(B,0)	0	0	0
world 1	Dunk1(P1,0)	1	1	1
	Dunk2(P1,0)	0	0	0
	Dunk1(P2,0)	0	0	0
	Dunk2(P2,0)	0	0	0
world 2	Dunk1(P1,0)	0	1	0
	Dunk2(P1,0)	0	0	0
	Dunk1(P2,0)	1	0	0
	Dunk2(P2,0)	0	0	0
world 1	defused(B,1)	1	1	1
world 2	defused(B,1)	1	1	0
	objective f	2	2	1

COMPUTATIONAL COMPLEXITY OF THE PROBLEM

Let us introduce complexity classes **P** and Σ_k**P**. Following (Baral et al. 2000), a decision problem is a problem of determining whether a given input w satisfies a certain property F (i.e., in set-theoretic terms, whether it belongs to the corresponding set $S = \{w|F(w)\}$). For every positive integer k, a problem belongs to the class Σ_k**P** if the formula $F(w)$ can be represented as:

$$\exists u_1 \forall u_2 \ldots F(u_1, u_2, \ldots, u_n, w),$$

where $F(u_1, u_2, \ldots, u_n, w)$ is a tractable property, and all k quantifiers run over words of tractable length (i.e., of length limited by some given polynomial of the length of the input).

Basing on above notation one can represent formula (1) for conformant planning as:

$$\exists \Delta\ \forall I\ \Pi\ (\Delta, \Pi(C, O, I, G)), \qquad (16)$$

where the initial state I can be potentially any state from states included in the set Σ. It follows that conformant planning is in $\mathbf{\Sigma_2 P}$. It is also a complete problem (Baral et al. 2000).

The complexity of the heuristic presented in the paper results from the size of LP problem, i.e. the number of variables and constraints for a given STRIPSIN problem (1). The number of variables depends on the number of conditions, actions and planning steps is:

$$p = w\,|C|\ (l+1) + w\ |O|\ l = p_1 + p_2, \qquad (17)$$

where:

– $p_1 = w|C|\ (l+1)$ – the number of variables corresponding to conditions,
– $p_2 = w|O|\ l$ – the number of variables corresponding to actions.

The number of constraints is:
- $w|C|$ equality constraints to define the initial state, since the number of constraints needed to define the initial state for each belief state is $|C|$,
- $|C|l$ equality constraints to define the change of variable values after performing the action,
- $|C|l$ inequality constraints to define actions preconditions,
- $2p$ inequality constraints for variable values <0,1>.

In a general case, for problems with the number of variables and constraints limited polynomially by the size of the planning problem, it can be shown (Galuszka 2011) that transformation of planning to LP takes time: $\boldsymbol{T}=\boldsymbol{O}(nl)$, where n is the size of the problem, $n = (w|C| + w|O|)$. If, additionally, it is assumed that the number of planning steps does not increase exponentially with the size of the problem, then transformation of planning to LP is polynomial with complexity $\boldsymbol{T} = \boldsymbol{O}(n^3)$. The heuristics of transformation of planning with incomplete information about initial state and determined effect of actions to LP has two properties:

a) one should introduce additional balancing variable for each condition in each planning step to avoid possible infeasibility of variable values,
b) given any feasible solution of LP problem x connected with planning problem $\Pi = \{C, O, \Sigma, G\}$, it is easy to check (in polynomial time) whether the solution corresponds to plan that solves Π.

From the property a) it follows that polynomial time, depending on the problem size, is needed to solve LP problems that represents incomplete planning: $\boldsymbol{T} = \boldsymbol{O}(n^3)$. From the property b) it follows that the heuristics is in NP.

CONCLUSION AND FUTURE WORK

Important planning problems are those where more than one agent interacts with the problem environment simultaneously. They arise in multi-agent and multi-robot environments. Additionally, it is assumed here that maximal number of actions applied to current problem state is r. It can occur when r agents act on the same problem state or one agent is able to perform r actions at a time. It should be noted that in real-life problems application of an action to a problem state does not always lead to expected effects. It is particularly important in cases where action outcomes are uncertain, as well, and when a condition that is determined can become undetermined. It leads to changes in constraints (12) and will be considered in future works.

Acknowledgments. This work has been supported by Institute of Automatic Control BK Grant 02/010/BK18/0102 (BK/200/Rau1/2018) in the year 2018 for the first (AG), third (JS) and fourth (AO) author and by Institute of Automatic Control BKM Grant for the second author (TG). The calculations were performed with the use of IT infrastructure of GeCONiI Upper Silesian Centre for Computational Science and Engineering (NCBiR grant no POIG.02.03.01-24-099/13).

REFERENCES

Baral Ch., Kreinovich V., Trejo R. (2000). Computational complexity of planning and approximate planning in the presence of incompleteness. *Artificial Intelligence*, 122, 241-.

Bonet B. (2010). Conformant plans and beyond: Principles and complexity. *Artificial Intelligence* 174 (2010) 245–269.

Bylander T. (1994). The Computational Complexity of Propositional STRIPS Planning. *Artificial Intelligence*, 69.

Bylander T. (1997). A Linear Programming Heuristic for Optimal Planning. In AAAI97/IAAI-97 Proceedings, 1997, 694—699.

Galuszka A., A. Swierniak. 2010. Planning in Multi-agent Environment Using Strips Representation and Non-cooperative Equilibrium Strategy. *Journal of Intelligent and Robotic Systems*, Vol. 58, Issue 3, 239 – 251.

Galuszka A. (2011). On transformation of STRIPS planning to linear programming. *Archives of Control Sciences*, Volume 21(LVII), No. 3, 227–251.

Galuszka A. (2013). Translation of Conditional Robot Action Planning to Linear Programming, Proc. 27th European Simulation and Modelling Conference - ESM'2013, October 23-25, 2013, Lancaster University, Lancaster, UK.

Galuszka Adam, Ilewicz Witold and Olczyk Adrian. (2015). On Translation of Conformant Action Planning to Linear Programming, 20th Int. Conf. on Methods and Models in Automation and Robotics (MMAR), ISBN: 978-1-4799-8701-6, pp. 353-357.

Ghooshchi Nina Ghanbari, Majid Namazi, M.A.Hakim Newton, Abdul Sattar. (2015). Transition Constraints for Parallel Planning. Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 3268-3274.

Kosindrdec N., J. Daengdej (2005). A Test Case Generation Process and Technique, Journal of Software Engineering 4(4), April 2010, pp. 265-287,.

Loucopoulos P., V. Karakostas (1995). System Requirements Engineeting, McGraw-Hill Inc.

Mich L., M. Franch, P. L. Novi Inverardi (2004). Market Research for Requirements Analysis Using Linguistic Tools, Requirements Engineering 9(1), January 2004, pp. 40-56.

Oglietti M. (2005). Understanding planning with incomplete information and sensing. Artificial Intelligence 164, 171–208.

Palacios H., Geffner H. (2009). Compiling Uncertainty Away in Conformant Planning Problems with Bounded Width. *Journal of Artificial Intelligence Research* 35, 623-675.

Requirements Working Group (2015). International Council on Systems Engineering (INCOSE), Guide for Writing Requirements, INCOSE-TP-2010-006-02, July 2015.

Rintanen J. (1999). Constructing Conditional Plans by a Theorem-Prover. *Journal of Artificial Intell.Research*, 10, 323-352.

Rosa T., S. Jimenez, R. Fuentetaja, D. Barrajo (2011). Scaling up Heuristic Planning with Relational Decision Trees. *Journal of Artificial Intelligence Research* , 40, 767-813.

Skrzypczyk K. (2010). Time optimal tracking a moving target by a mobile vehicle - game theoretical approach, *Przegląd Elektrotechniczny (Electrical Review)*, R.86 NR 3/2010 s. 211-.

Smith D.E., Weld D.S. (1998). Conformant Graphplan. Proc. 15th National Conf. on AI.

Son Tran Cao, Phan Huy Tu, Michael Gelfond, and A. Ricardo Morales. 2005. Conformant planning for domains with constraints: a new approach. In Proceedings of the 20th national conference on Artificial intelligence - Volume 3 (AAAI'05), Anthony Cohn (Ed.), Vol. 3. AAAI Press 1211-1216.

Weld D.S., Anderson C.R., Smith D.E. (1998). Extending Graphplan to Handle Uncertainty & Sensing Actions. Proc. 15th National Conf. on AI, 897-904.Author, F.: Article title. Journal 2(5), 99–110 (2016).

MULTI-OBJECTIVE OPTIMIZATION DRIVEN CONSTRUCTION OF UNIFORM PRIORS FOR LIKELIHOOD-FREE PARAMETER INFERENCE

Prashant Singh and Andreas Hellander
Division of Scientific Computing
Department of Information Technology
Uppsala University
Box 337, SE-751 05
Uppsala, Sweden.
email: prashant.singh@it.uu.se

KEYWORDS
Parameter Inference, Approximate Bayesian Computation, Likelihood-Free Inference, Multi-Objective Optimization

ABSTRACT

Likelihood-free parameter inference methods such as approximate Bayesian computation often assume a uniform prior within a sufficiently large range in absence of problem-specific knowledge. This potentially leads to large high-dimensional search spaces. A low-fidelity variant of the simulator can be used for detailed preliminary analysis of the search space with the aim of reducing the range of the uniform prior. This paper explores multi-objective optimization for analysis and reduction of uniform prior configurations towards achieving high quality inference. The inference problem is formulated as a multi-objective minimization problem, the solution of which leads to a reduced search space for the uniform prior. The proposed approach is demonstrated on a complex parameter inference problem from systems biology where it achieves a substantial search space reduction.

INTRODUCTION

Simulation models are routinely used for deep understanding of natural phenomena and complex processes. Simulator parameters are often tuned to agree with observed or experimental data, so that the simulator can be used to perform exploratory analysis, studies on parameter sensitivity, optimization, etc. One of the first approaches to solve such parameter inference problems was maximum likelihood estimation (MLE) (Beerli 2005) wherein the likelihood of the simulator parameters fitting the data is maximized. However, for a vast majority of real-world problems, the likelihood function is either unavailable due to problem complexity, or computationally too expensive to evaluate. This has motivated research towards likelihood-free parameter inference.

The most widely used likelihood-free inference method is approximate Bayesian computation (ABC) (Pritchard et al. 1999). The proposed approach is presented in the context of ABC, but is equally applicable for other inference methods that assume a uniform prior. ABC involves approximation of the likelihood function with the aid of a rejection sampling method. Samples are generated according to a specified prior distribution, and subsequently simulated. A sample is accepted if the simulated response is within a specified tolerance limit of the observed data set, as measured by a combination of a specified summary statistics and a distance function. The inferred parameters are reported as the mean of accepted sample parameter values. ABC and its derivatives have proven to be robust and have delivered reliable performance across disciplines (Lintusaari et al. 2017). However, setting optimal values of ABC hyperparameters such as the tolerance threshold, choice of the prior function, the distance function and summary statistics, etc. is challenging. In absence of optimal hyperparameters, the convergence towards inferred parameters can be slow.

The problem structure of an inference problem includes specification of variables such as the parameter search space, choice of the distance function and summary statistics. In cases where access to a low-fidelity fast-to-evaluate variant of the simulator is available, it is possible to intelligently search for optimal values of the problem variables or hyperparameters. In absence of computationally inexpensive simulators, the practitioner has to rely on domain knowledge and past experience with similar problems. Examples of low-fidelity computationally inexpensive simulator variants include coarse mesh-based models, simulations involving fewer trajectories of a stochastic process (Gillespie 1977) or even surrogate models of computationally expensive simulators (Singh and Hellander 2017).

Such fast approximations can be used to perform trial ABC runs to gain insight into well-performing hyperparameter values. This work considers the problem of identifying one of the hyperparameters - the prior function that specifies the parameter search space. In absence of any problem-specific information, as is often the case in black-box likelihood-free inference, a uniform prior is assumed. This typically translates into selection of a relatively large, continuous range of values of each parameter by the practitioner. For large-scale inference problems involving tens of parameters, the resulting search space is substantial. The enormity of the search space makes parameter inference a daunting, time-consuming endeavor. This work proposes a novel method

that utilizes access to a fast low-fidelity simulator to substantially reduce the parameter search space corresponding to a uniform prior.

The search space reduction problem is formulated as a multi-objective optimization problem. The distance value corresponding to each summary statistic is treated as an objective to be minimized. The proposed multi-objective formulation is used to arrive at a Pareto set of solutions depicting trade-offs (in the form of summary statistic distances) that provides useful insight into the performance of summary statistics and the parameter search space. We empirically demonstrate the utility of the proposed approach herein by conducting parameter search space reduction of a high-dimensional complex parameter inference problem from the systems biology literature.

PARAMETER INFERENCE AS OPTIMIZATION

The following text formally introduces the parameter inference problem, and motivates a formulation in the context of multi-objective optimization. Consider a simulator function f parameterized by variables $\boldsymbol{\theta}$, and an observed data set consisting of values $\mathbf{y}_{obs}$. Assuming a uniform prior, each parameter θ_i comprising $\boldsymbol{\theta}$ has a corresponding search range $[min_{\theta_i}, max_{\theta_i}]$. The parameter inference task is to estimate the value of parameters $\boldsymbol{\theta}$ that when evaluated using the simulator as $\mathbf{y}_{sim} = f(\boldsymbol{\theta})$, result in responses $\mathbf{y}_{sim} = \mathbf{y}_{obs}$. This equality condition is too stringent for practical purposes, and is typically replaced by a more relaxed condition $d(\mathbf{y}_{sim}, \mathbf{y}_{obs}) \leq \tau$, where d is a chosen distance function operating over specified *summary statistics* and τ is the tolerance bound. Intuitively, one is willing to accept $\mathbf{y}_{sim}$ and $\mathbf{y}_{obs}$ to be sufficiently similar if the computed distance between them is within the tolerance bound.

This tolerance bound τ not only accounts for stochasticity and general noise in the considered processes, but also reflects the trade-off between accuracy and inference speed. The smaller the value of τ, the higher the desired inference accuracy is. In practice, smaller values of τ lead to longer inference times as well (Lintusaari et al. 2017).

As simulated responses are often complex and time-varying, they are often reduced to low-dimensional representations or features, known as summary statistics. Popular summary statistics include statistical moments such as mean, minima, maxima, in addition to quantities such as Fourier transform coefficients, etc. Distance(s) are often computed in terms of one or more summary statistics during the inference process. Complex parameter inference problems may involve tens to hundreds of parameters, leading to very high-dimensional search spaces represented by the prior $p(\boldsymbol{\theta})$. This renders the inference process computationally expensive and impractical. The following text introduces multi-objective optimization as means to reduce the search space, and enable faster and accurate parameter inference.

Let $\mathbf{d} = (d_1, ..., d_k)$ be a vector of distance function values calculated over k summary statistics $S = (s_1, ..., s_k)$ as $d_i = d(s_i(f(\boldsymbol{\theta})), s_i(\mathbf{y}_{obs}))$ using a chosen distance function d. The structure of the multi-objective optimization problem considered in this work is:

$$\begin{aligned} &\text{min. } \mathbf{d} = (d_1, ..., d_k), \\ &\text{s. t. } \boldsymbol{\theta} \sim p(\boldsymbol{\theta}). \end{aligned}$$

The solution of such a problem using evolutionary multi-objective optimization (Zhou et al. 2011) or Bayesian multi-objective optimization (Singh et al. 2014, Couckuyt et al. 2014) will result in a Pareto set of solutions showing trade-offs between various summary statistic distances. The Pareto set can be analyzed by the practitioner to identify regions where solutions corresponding to smaller distances lie. Consequently, the practitioner can also get a feel of the corresponding parameter space resulting in smaller distances, and accordingly adopt the hyperparameters, including the range of samples $\boldsymbol{\theta}$ that the prior can generate. For example, in case of the uniform prior, the range of each parameter can be made tighter by identifying 'good' regions using the Pareto set. This can lead to substantial savings in computational time and greatly improve the quality of inference achieved with the eventual inference run using the high-fidelity simulator.

Figure 1 depicts the multi-objective optimization based search space reduction framework. The Pareto set of solutions is analyzed by the practitioner to identify a small set of *good* solutions $\mathbf{y}'_{sim} \subset \mathbf{y}_{sim}$. In general, goodness of solutions should correspond to smaller distance values represented by the objectives. In absence of any domain-specific knowledge or preset priorities between objectives, the 'knee' of the Pareto set of solutions signifies equal compromise and trade-off between the objects and is desirable. This scenario is depicted in Figure 1 where the identified *good* set of solutions $\mathbf{y}'_{sim}$ is encircled. The practitioner then proceeds to extract the simulator inputs or parameter values $\boldsymbol{\theta}$ that resulted in simulated values $\mathbf{y}'_{sim}$. The boundary of the reduced search space is then obtained as the extrema of each parameter θ_i comprising $\mathbf{y}'_{sim}$. The following text briefly discusses multi-objective optimization algorithms used in this work.

Multi-Objective Optimization Algorithms

A popular approach for the solution of black-box multi-objective optimization problems is using multi-objective evolutionary algorithms (MOEAs) (Zhou et al. 2011). MOEAs are a flexible general choice and have proven to be effective across diverse problem domains (Jaimes and Coello Coello 2017). MOEAs start with an initial *population* of candidate solutions (often selected randomly) that are iteratively refined using the *crossover* and *mutation* operators. The operators retain their philosophical meaning from evolutionary biology and encourage creation of offspring solutions in each iteration with good fitness function (objective function) values, while retaining diversity. The progressive evolution of 'good' solutions results in the final Pareto set of solutions.

The MOEAs evaluated in this work include the non-dominated sorting genetic algorithm II (NSGA II) (Deb et al.

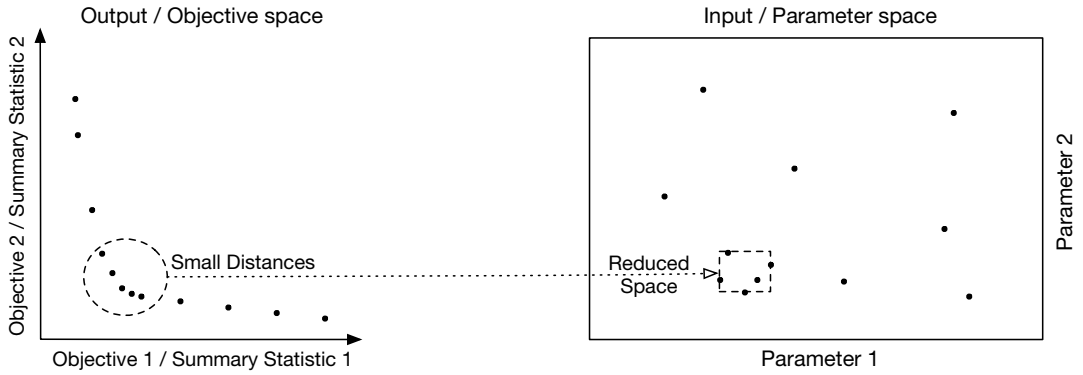


Figure 1: Multi-objective optimization driven search space reduction for parameter inference. The Pareto set of solutions can be used to obtain a substantially reduced search space corresponding to solutions with small distances.

2002), the strength Pareto evolutionary algorithm 2 (SPEA2) (Zitzler et al. 2001), and multi-objective evolutionary algorithm based on decomposition (MOEA/D) (Zhang and Li 2007). A detailed discussion of the algorithms is out of scope of this work, and the reader is referred to Zhou et al. (2011) for a general overview, and to the respective paper of each MOEA for deeper details.

EXPERIMENTS

To demonstrate the approach, we consider a stochastic model of the dynamics of proteins involved in a gene regulatory network (Vilar et al. 2002). The model is a continuous-time discrete space Markov process, where chemical reactions between the involved proteins are simulated as transitions, and where the transition rates depend on kinetic rate parameters. This type of model is commonly employed in computational systems biology to study the behavior of biochemical pathways in the presence of intrinsic molecular noise (Elowitz et al. 2002). Statistically exact samples of the stochastic process can be generated with the Stochastic Simulation Algorithm (SSA) (Gillespie 1977). Exact simulations are computationally expensive for the so called stiff system, and being a stochastic model, large ensembles of independent realizations are usually needed to estimate properties of the system accurately. A large number of approximate simulation methods have been developed, and there is a mathematically well-motivated hierarchy of cheaper-to-simulate low-fidelity models, including ordinary differential equations, diffusion approximations, moment expansions and stochastic differential equations, for an overview see (Gillespie et al. 2013).

The network we study here is an oscillatory model of a circadian rhythm. It has been used extensively as a model problem for development of efficient algorithms and software due to its stiffness. It consists of 9 species undergoing 18 reactions, with 15 reaction rate constants. Let $\mathscr{S} = \{D_A, D_A^*, M_A, D_R, D_R^*, M_R, C, A, R\}$ be the set of species with initial copy numbers $\{1,0,0,1,0,0,10,10,10\}$ respectively. The model involves the reactions listed in Eq. 1,

$$
\begin{aligned}
&sD_A : D_A^* \xrightarrow{\theta_A} D_A, && sA_3 : D_R^* \xrightarrow{\theta_A} D_R^*, A,\\
&sD_A^* : D_A, A \xrightarrow{\gamma_A} D_A^*, && aA : A \xrightarrow{\delta_A} \phi,\\
&sD_R : D_R^* \xrightarrow{\theta_R} D_R, && sC : A, R \xrightarrow{\gamma_C} C,\\
&sD_R^* : D_R, A \xrightarrow{\gamma_R} D_R^*, && sMR_1 : D_R^* \xrightarrow{\alpha_{R^*}} D_R^*, M_R,\\
&sMA_1 : D_A^* \xrightarrow{\alpha_A^*} D_A^*, M_A, && sMR_2 : D_R \xrightarrow{\alpha_R} D_R, M_R,\\
&sMA_2 : D_A \xrightarrow{\alpha_A} D_A, M_A, && aMR : M_R \xrightarrow{\delta_{MR}} \phi,\\
&aM_A : M_A \xrightarrow{\delta_{MA}} \phi, && sR_1 : M_R \xrightarrow{\beta_R} M_R, R,\\
&sA_1 : M_A \xrightarrow{\beta_A} A, M_A, && aR : R \xrightarrow{\delta_R} \phi,\\
&sA_2 : D_A^* \xrightarrow{\theta_A} D_A^*, A, && sR_2 : C \xrightarrow{\delta_A} R.
\end{aligned}
\tag{1}
$$

Parameter inference is performed within the search space corresponding to Eq. 2,

$$
\begin{aligned}
&\alpha_A \in [30,70], && \delta_A \in [0,2],\\
&\alpha_A^* \in [200,600], && \delta_R \in [0,0.5],\\
&\alpha_R \in [0,1], && \gamma_A \in [0.5,1.5],\\
&\alpha_R^* \in [30,70], && \gamma_R \in [0.5,1.5],\\
&\beta_A \in [30,70], && \gamma_C \in [1,3],\\
&\beta_R \in [1,10], && \theta_a \in [30,70],\\
&\delta_{MA} \in [1,12], && \theta_r \in [80,120].\\
&\delta_{MR} \in [0,1], &&
\end{aligned}
\tag{2}
$$

Experimental Setup

The Platypus multi-objective optimization Python library (Hadka 2015) implements all three algorithms (NSGA II, SPEA2, MOEA/D) and is used for experiments in this paper. The number of iterations or generations allowed for each algorithm is 400 with the population size being 100 individuals. The true parameters used to generate the observed data set are $\{50, 500, 0.01, 50, 50, 5, 10, 0.5, 1, 0.2, 1, 1, 2, 50, 100\}$. The

summary statistics used in the experiments are the 77^{th} Fourier coefficient of the one-dimensional discrete Fourier transform obtained using the Fast Fourier Transform (FFT) algorithm, and mass quantile index i where 0.9% of the values of the input time series lie to the left of i. These summary statistics are chosen according to a previous extensive study of over 500 summary statistics applied to the considered problem (Singh and Hellander 2018).

Results

Table 1 lists the values of the hypervolume indicator which is a well-established metric to evaluate performance of multi-objective optimization algorithms. The hypervolume indicator measures the span covered by the solutions relative to a reference point. As the objective function values used in this work correspond to distances measured using normalized summary statistics $\in [0,1]$, the reference point used for computation consists of $(0,0)$ as the lower bound and $(1,1)$ as the upper bound for the two objectives.

Table 1: Mean values of the hypervolume indicator and execution time t_{exec} over 10 runs. Higher values are preferable.

Algorithm	Hypervolume Indicator	t_{exec} (s)
NSGA II	0.9975 ± 0.0014	296.7036
SPEA2	0.9998 ± 0	371.5965
MOEA/D	0.9996 ± 0.0001	456.3917

It can be observed that all three algorithms perform well and are comparable in terms of the hypervolume indicator. The low standard deviation of the values of the hypervolume indicator points towards robust and consistent optimization.
The resulting Pareto sets can be seen in Fig. 2. It can be observed that the solutions found using SPEA2 are most exploration oriented, while MOEA/D has highly exploitation-based behavior. NSGA II lies in between the two and offers a balance of exploration and exploitation. The mean running times of all algorithms are very practical with NSGA II being the quickest and MOEA/D the slowest. It is interesting to note that the mass quantile objective was easier to satisfy as compared to the Fourier coefficient. NSGA II and SPEA2 explored a wider range of solutions in terms of the Fourier coefficient, while MOEA/D concentrated solely on the region where both objectives had smaller values.
Although Fig. 2 provides an indication of the nature of solutions obtained using the three algorithms, 400 iterations can be too few for the evolutionary operators to realize their full potential. Therefore, a more thorough optimization process involving 5000 iterations is performed using the NSGA II algorithm, which proved to be a balanced choice during the preliminary comparison. The optimization process took 5184.5663 seconds to complete. The resulting Pareto set is shown in Fig. 3a, and a magnified view of the Pareto-optimal solutions lying close to the origin in the objective space is shown in Fig. 3b. The magnified view is arrived at simply by zooming in towards the knee of the Pareto set and constitutes balanced trade-offs between the objectives. Based on the parameter values corresponding to the solutions seen in Fig. 3, it is possible to define a reduced search space represented by Eq. 3,

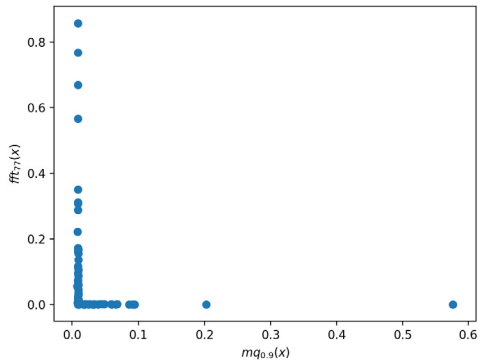
$$\begin{aligned}
\alpha_A &\in [37.74, 66.89], & \delta_A &\in [0.2696, 1.97],\\
\alpha_A^* &\in [319.37, 574.23], & \delta_R &\in [0.04, 0.50],\\
\alpha_R &\in [0.03, 0.99], & \gamma_A &\in [0.85, 1.47],\\
\alpha_R^* &\in [45.12, 69.46], & \gamma_R &\in [0.51, 1.40],\\
\beta_A &\in [39.92, 69.98], & \gamma_C &\in [1.02, 2.97],\\
\beta_R &\in [4.15, 9.32], & \theta_a &\in [30.06, 68.91],\\
\delta_{MA} &\in [1.16, 9.98], & \theta_r &\in [82.70, 119.79].\\
\delta_{MR} &\in [0.08, 0.99],
\end{aligned} \tag{3}$$

The hypervolume of the reduced search space is approximately 1.4776% of the original search space, resulting in a substantial reduction of 98.5224%. This allows for improved inference quality on the full-fidelity simulator using the same ABC settings, except for the span of search spaces. Table 2 compares the root relative squared error (RRSE) of two configurations - the reduced search space (Red.) with $\tau = 0.05$, and the original search space (Orig.) with $\tau = 0.03$ and $\tau = 0.05$. Each value is the mean of 10 runs. The ABC settings included the requirement of 50 accepted samples and distance function values normalized to lie in $[0,1]$.

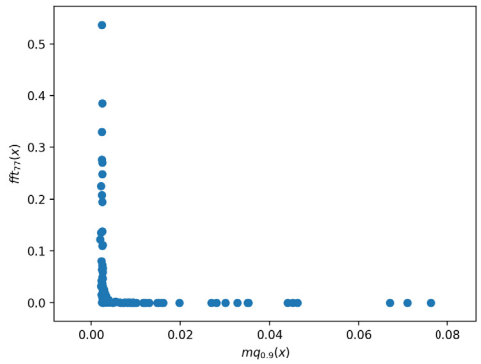
Table 2: RRSE in inferred parameters, the mean distance of accepted samples d_μ, and the number of simulations conducted by ABC parameter inference. Smaller values are more desirable.

Search Space	RRSE	d_μ	# simulations
(Orig., $\tau = 0.03$)	0.29 ± 0.04	**0.019**	392.7 ± 50.19
(Orig., $\tau = 0.05$)	0.28 ± 0.03	0.027	$\mathbf{228.6 \pm 25.37}$
(Red., $\tau = 0.05$)	$\mathbf{0.15 \pm 0.02}$	0.030	248 ± 37.53

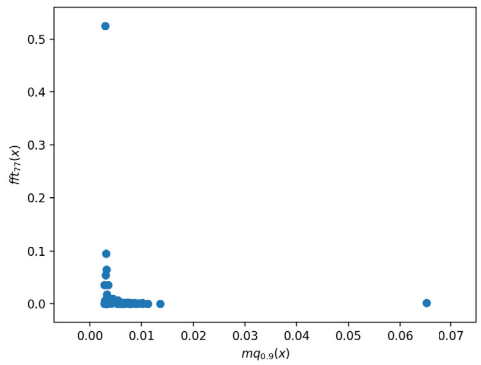
The value $\tau = 0.03$ was chosen by calculating the mean distance of accepted samples from executing the ABC configuration corresponding to the reduced search space, and τ being 0.05. Intuitively, this should result in similar inference quality albeit with the search in the original space taking longer. It is interesting to note that, even with a tighter tolerance bound of $\tau = 0.03$, search in the original space resulted in worse inference quality with respect to RRSE in inferred parameters. This can be attributed to the fact that there can be different regions in the summary statistic space with behavior satisfying the specified tolerance bound. Hence, different values of input parameters $\boldsymbol{\theta}$ can produce same or similar (taking into account stochastic behavior) values of summary statistics upon simulation. The values of d_μ (mean distance of accepted samples) confirm this assertion with $\tau = 0.03$ resulting in smallest observed mean distance of 0.019 in summary statistics. However, the corresponding RRSE is the highest with a value of 0.29. This shows that inference quality (as measured by the error in inferred parameters) does not



(a) Pareto set found using NSGA II.

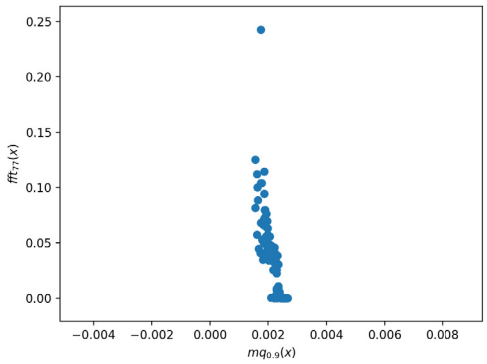


(b) Pareto set found using SPEA2.

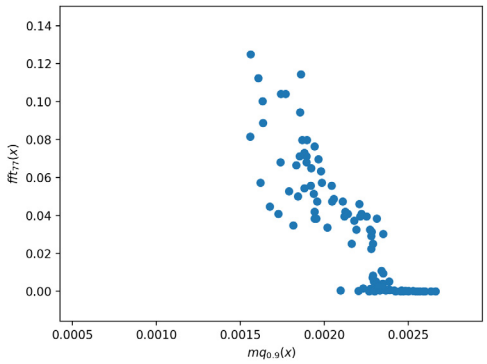


(c) Pareto set found using MOEA/D.

Figure 2: The resulting Pareto sets found using the three MOEAs.



(a) Pareto set found using NSGA II with 5000 iterations allowed.



(b) Magnified view of the region of interest in Fig. 3a.

Figure 3: NSGA II Results: Full and magnified views.

always correlate directly with observed distance; i.e., smaller distances do not always imply good inference quality. Within a reduced search space, the likelihood of exploring such sub-optimal regions decreases and this translates into better inference quality. This is reflected in the results corresponding to the reduced search space with $d_\mu = 0.03$ being highest among all configurations, but RRSE = 0.15 being the lowest by a margin. The number of simulations taken by each configuration to achieve 50 accepted samples are as expected, with the tighter tolerance bound $\tau = 0.03$ requiring more simulations.

For complex parameter inference problems involving a large number of parameters, the impact of search space reduction can be substantial. However, the MOEA must be given sufficient number of iterations in order to fully realize the reduction potential. For instance, the true value of the parameter α_R is 0.01, which unfortunately lies outside the reduced search space as the new range of α_R is $[0.03, 0.99]$. In order to avoid such a scenario the MOEA must be allowed to evolve for a sufficiently large number of iterations (typically of the order of few thousands). As the computationally cheap low-fidelity simulator takes fractions of a second, allowing the MOEA to evolve for a few thousand iterations is practical. However, MOEA-driven preprocessing is not realistic for simulators that are computationally expensive, with each simulation taking multiple seconds. For example, each simulation using the full-fidelity simulator used herein takes approximately 2.65 seconds to complete. Invoking the simulator 10000 times will involve 7+ hours of computation, which is not practical for pre-processing.

Discussion

It can be argued that search space reduction is inherent in ABC rejection sampling, as the prior is continually updated to sample more intensely in regions corresponding to low distances. However, for large scale parameter inference problems (involving ten or more parameters), the search towards such regions can be slow and arduous in absence of an informative prior (as is often the case). The proposed approach is useful in such problem settings and complements the inference process using ABC and related methods.
The proposed approach is dependent upon the availability of computationally inexpensive simulators. In absence of a fast simulator, data efficient optimization methods such as Bayesian optimization can be used in lieu of MOEAs (Singh et al. 2014).
Future work includes investigating incorporation of the proposed MOEA-based framework directly within likelihood-free inference methods such as ABC. Different approaches and metrics to quantify inference quality will also be explored, with the goal of arriving at better formulations of the objective functions.

CONCLUSION

A novel approach for analysis of summary statistics and reduction of search space for parameter inference problems was presented in this paper. The approach is based on multi-objective optimization and utilizes access to a computationally inexpensive variant of the simulator. Empirical evaluation on a complex parameter inference problem from systems biology demonstrated the efficacy of the proposed approach. Substantial search space reduction of over 98% and improvement in inference quality was achieved as a result of preliminary analysis performed using the proposed approach.

ACKNOWLEDGEMENTS

This work was funded by the Göran Gustafsson foundation and the eSSENCE strategic collaboration on eScience.

REFERENCES

Beerli P., 2005. *Comparison of Bayesian and maximum-likelihood inference of population genetic parameters*. *Bioinformatics*, 22, no. 3, 341–345.

Couckuyt I.; Deschrijver D.; and Dhaene T., 2014. *Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization*. *Journal of Global Optimization*, 60, no. 3, 575–594.

Deb K.; Pratap A.; Agarwal S.; and Meyarivan T., 2002. *A fast and elitist multiobjective genetic algorithm: NSGA-II*. *IEEE transactions on evolutionary computation*, 6, no. 2, 182–197.

Elowitz M.B.; Levine A.J.; Siggia E.D.; and Swain P.S., 2002. *Stochastic gene expression in a single cell*. *Science*, 297, no. 5584, 1183–1186.

Gillespie D.T., 1977. *Exact stochastic simulation of coupled chemical reactions*. *The journal of physical chemistry*, 81, no. 25, 2340–2361.

Gillespie D.T.; Hellander A.; and Petzold L.R., 2013. *Perspective: Stochastic algorithms for chemical kinetics*. *The Journal of Chemical Physics*, 138, no. 17, 170901.

Hadka D., 2015. *Platypus - Multiobjective Optimization in Python*. `https://github.com/Project-Platypus/Platypus`.

Jaimes A.L. and Coello Coello C.A., 2017. *Multi-objective evolutionary algorithms: A review of the state-of-the-art and some of their applications in chemical engineering*. In *MULTI-OBJECTIVE OPTIMIZATION: Techniques and Application in Chemical Engineering*, World Scientific. 63–92.

Lintusaari J.; Gutmann M.U.; Dutta R.; Kaski S.; and Corander J., 2017. *Fundamentals and recent developments in approximate Bayesian computation*. *Systematic biology*, 66, no. 1, e66–e82.

Pritchard J.K.; Seielstad M.T.; Perez-Lezaun A.; and Feldman M.W., 1999. *Population growth of human Y chromosomes: a study of Y chromosome microsatellites. Molecular biology and evolution*, 16, no. 12, 1791–1798.

Singh P.; Couckuyt I.; Ferranti F.; and Dhaene T., 2014. *A constrained multi-objective surrogate-based optimization algorithm*. In *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 3080–3087.

Singh P. and Hellander A., 2017. *Surrogate assisted model reduction for stochastic biochemical reaction networks*. In *Proceedings of the 2017 Winter Simulation Conference (WSC)*. IEEE, 1773–1783.

Singh P. and Hellander A., 2018. *Multi-Statistic Approximate Bayesian Computation with Multi-Armed Bandits*. *arXiv preprint arXiv:180508647*.

Vilar J.M.; Kueh H.Y.; Barkai N.; and Leibler S., 2002. *Mechanisms of noise-resistance in genetic oscillators*. *Proceedings of the National Academy of Sciences*, 99, no. 9, 5988–5992.

Zhang Q. and Li H., 2007. *MOEA/D: A multiobjective evolutionary algorithm based on decomposition*. *IEEE Transactions on evolutionary computation*, 11, no. 6, 712–731.

Zhou A.; Qu B.Y.; Li H.; Zhao S.Z.; Suganthan P.N.; and Zhang Q., 2011. *Multiobjective evolutionary algorithms: A survey of the state of the art*. *Swarm and Evolutionary Computation*, 1, no. 1, 32–49.

Zitzler E.; Laumanns M.; and Thiele L., 2001. *SPEA2: Improving the strength Pareto evolutionary algorithm*. *TIK-report*, 103.

A SIMULATION-BASED TECHNIQUE FOR CONTINUOUS-SPACE EMBEDDING OF DISCRETE-PARAMETER QUEUEING SYSTEMS

Neha Karanjkar*
Robert Bosch Centre for Cyber-Physical Systems, IISc Bangalore

Madhav Desai
Department of Electrical Engineering, IIT Bombay

Shalabh Bhatnagar
Department of Computer Science and Automation, IISc Bangalore

KEYWORDS
Simulation Optimization, Queueing, Interpolation

ABSTRACT

This paper presents a simulation-based technique for embedding the discrete-valued parameters in queueing systems (such as buffer capacities) into a continuous space. The significance of this technique is that it enables the direct application of existing descent-based continuous optimizers for solving simulation optimization problems efficiently over a discrete parameter set. The embedding technique is based on a randomization of the simulation model itself, and is applicable when the objective function is a long-run average measure. Unlike spatial interpolation, the computational cost of this embedding is independent of the number of parameters in the system, making the approach well-suited to high-dimensional problems. We present a theoretical basis for this embedding technique and demonstrate its utility in the optimization of discrete-time queueing systems.

INTRODUCTION

The use of simulation is often necessary in the optimization of complex real-life queueing networks. Such queueing networks typically have discrete valued parameters such as queue capacities, the number of servers and service-times in slotted-time queues. More concretely, consider a queueing system with a parameter set $X = (x_1, x_2, \ldots, x_n)$ where each x_i can take integer values between some fixed bounds. The set of all possible values that X can take is the n-dimensional, discrete parameter space Ω_D. Let $f : \Omega_D \rightarrow \mathbb{R}$ be some cost/performance measure of the system that we wish to optimize. In many problems, f may be composed of long-run average measures such as average throughput, average waiting time per customer or blocking probabilities. An analytical expression for f is rarely available and given X, $f(X)$ can only be measured using a simulation of the system. The measurements are noisy as each simulation run has finite length. We are motivated by the problem of finding an $X^* \in \Omega_D$ that minimizes $f(X)$. This is a *Discrete-Parameter Simulation Optimization* (DPSO) problem. The problem is often difficult as the number of parameters can be very large and each function evaluation is computationally expensive. Further, most discrete-space search methods do not scale well with the number of parameters.

For small parameter sets, ranking and selection-based procedures such as Optimal Computing Budget Allocation (Chen and Lee (2010)) have been effectively applied. When the number of parameters is large, an exhaustive evaluation of all design points is infeasible. The goal then is to find the best possible solution within a finite computational budget, rather than the global optimum. In such a case, randomized search techniques such as simulated annealing and genetic algorithms or heuristic-based local search methods such as Tabu search have been employed. A detailed survey of simulation optimization approaches can be found in Swisher et al. (2000). Discrete-space variants of continuous optimizers such as Simultaneous Perturbation Stochastic Approximation (SPSA) have also been proposed wherein the parameter estimate is projected back to the discrete space at each iteration (see Whitney et al. (2001) and Bhatnagar and Kowshik (2005)). In all of the above methods, the objective function evaluation is limited strictly to points in the original discrete domain.

A Continuous-space Embedding Approach:

If the discrete parameter space can be *embedded* into a larger continuous space by using some form of interpolation, the optimization problem can be solved by directly applying descent-based continuous-space methods. Such continuous optimizers often scale well with respect to the number of parameters, in comparison to discrete-space methods. This is because gradient information can be utilized at each step to converge rapidly to local minima. To search for a global optimum, random multi-starts can be used. The solutions thus found in the continuous domain are projected back to the discrete domain (for example, using nearest-neighbour rounding). Let $\Omega_C \subset \mathbb{R}^n$ denote the convex hull of the original discrete space Ω_D. An embedding of the discrete parameter space into a continuous one is essentially an interpolation (say $\hat{f}$) of f defined over Ω_C. If $\hat{f}$ can be constructed in a computationally efficient manner and has a suitable structure (that is, $\hat{f}$ is continuous, piece-wise smooth, has few local minima) then continuous optimizers applied directly over $\hat{f}$ can be expected to perform well.

Such an embedding-based approach was shown to be effective in the optimization of queueing and inventory sys-

*Corresponding author. Email: neha.karanjkar@rbccps.org

tems in the past (see Lim (2012) and Wang and Schmeiser (2008)). However the embedding reported in existing literature was achieved via spatial interpolation (for instance, using piece-wise simplex interpolation) which requires multiple simulations to be performed at each parameter point. More concretely, given some point $Y \in \Omega_C$, and a set of points $X^1, X^2, \ldots, X^p \in \Omega_D$ in the neighbourhood of Y, the interpolated objective value $\hat{f}(Y)$ can be computed as a weighted average of the values $f(X^1), f(X^2), \ldots, f(X^p)$, each of which is obtained via a single simulation. Thus p simulations are required to estimate the interpolated value. For linear interpolation, $p = 2^n$ and for piece-wise simplex interpolation, $p \geq n + 1$. Thus, spatial interpolation as a means of embedding is computationally expensive.

Instead, this paper presents an embedding technique which requires a single simulation to measure the interpolated value at a given point in the continuous domain (irrespective of the number of parameters). The technique is based on a randomization of the simulation model and is applicable when the objective f is composed of long-run average measures. To perform the embedding, each parameter in the model is perturbed periodically and assigned values of a discrete random variable, instead of a fixed constant value over a single simulation run. Now, the distribution of this random variable can be varied continuously, even though the model parameter itself is discrete-valued. The value of the objective function measured using a single, long simulation of this randomized model directly gives us the interpolated objective value. In essence, the technique relies on an averaging in *time*, in contrast to spatial interpolation methods which perform an averaging over the parameter space. We describe the embedding technique in more detail in the following section. In this paper, we present a theoretical justification for the randomization-based embedding technique and describe its application to discrete-time queueing systems.

Related Work and Our Contributions

The use of randomization for embedding discrete-valued parameters in a simulation model was reported by Karanjkar and Desai (2015) for the design optimization of multi-core systems, however without a theoretical justification. A theoretical basis for such a technique was introduced by Bhatnagar et al. (2011) in the context of two specific algorithms for solving the DPSO problem. This work proposed variants of two continuous optimizers wherein the parameter estimate at each iteration is projected back to the discrete space probabilistically (rather than in a deterministic manner) and showed that this essentially produces a continuous embedding of the underlying discrete-parameter process. The work focused on the optimization algorithms and the embedding technique itself was not explored in depth.

The focus of the current paper is on the randomization-based embedding technique itself. We present a general randomization scheme and prove that it produces continuous interpolations of the objective. The proof extends (Bhatnagar et al. 2011; Lemma 3) by relaxing the assumption on the ergodicity of the constituent Markov chains, making the analysis applicable to a wider set of parameters and systems (including systems with transient and/or periodic states), such as the examples considered in this paper. We then describe the application of the embedding technique to discrete-time (slotted) queues for embedding queue-capacity, number of servers and service-time parameters into a continuous space. Such queues are of importance in several applications such as communication networks, manufacturing lines and transportation systems (see Alfa (2015)). To demonstrate the utility of the embedding technique, we consider the optimization of a queueing network with 7 parameters. We observe that a randomization of the simulation model produces continuous, smooth embeddings of the objective and two continuous optimizers applied directly over this embedding perform favourably in comparison to a direct discrete-space search method. However the focus of the current paper is on the embedding technique itself, rather than on specific optimization methods. In-fact, once an embedding has been achieved, a rich set of existing continuous optimizers become applicable to the original discrete-parameter problem. A detailed performance comparison between discrete-space methods and continuous optimizers applied over an embedding can be the topic for future research and is beyond the scope of the current paper. In summary, this paper introduces the randomization-based embedding technique as a useful tool in the optimization of queueing networks over discrete parameters. The simulation models and scripts used in this work are available in an online repository (see Karanjkar (2017)).

RANDOMIZATION-BASED EMBEDDING

Consider a system with n integer-valued parameters and let $X = (x_1, x_2, \ldots, x_n)$ denote the parameter vector, where each x_i can take values from some finite set D_i consisting of successive integers. The set of all possible values that X can take is the n-dimensional discrete parameter space Ω_D which, in this case is the Cartesian product $\Omega_D = \prod_{i=1}^{n} D_i$. Given a point $X \in \Omega_D$, our simulation model allows us to measure the value of some long-run average objective $f(X)$. Let Ω_C denote the convex hull of Ω_D. We wish to obtain the interpolation $\hat{f} : \Omega_C \rightarrow \mathbb{R}$ of f, and measure its value at a given point $Y \in \Omega_C$. Let $Y = (y_1, y_2, \ldots, y_n)$. Thus the i^{th} parameter needs to be assigned a value $y_i \in \mathbb{R}$ in the embedded model.

To obtain $\hat{f}(Y)$, we construct a randomized version of the model where the value of each parameter in the model is perturbed periodically (for example, at the beginning of each time-slot) and assigned values of a discrete random variable which we denote as γ. For each $i \in \{1, \ldots, n\}$, the i^{th} parameter in the model is assigned values of the random variable $\gamma_i(y_i)$. The random variable γ_i is chosen in such a way that its moments can be made to vary continuously with respect to the parameter y_i, and $\gamma_i(y_i) = y_i$ with probability 1 whenever $y_i \in D_i$. The simplest example of such a random

variable is:

$$\gamma(y) = \begin{cases} \lfloor y \rfloor & \text{with probability } \alpha(y) \\ \lceil y \rceil & \text{with probability } 1 - \alpha(y) \end{cases} \quad (1)$$
$$\text{where } \alpha(y) = \lceil y \rceil - y$$

For instance, if $y_i = 10.3$, then at each time-slot the parameter will take the value 11 with probability 0.3 and the value 10 with probability 0.7 so that its average value over a single simulation run would be 10.3. All parameters in the model are embedded simultaneously and assigned values of independent random variables $\gamma_1(y_1), \gamma_2(y_2), \ldots \gamma_n(y_n)$. Let $\hat{f}$ be the long-run average measure obtained by simulating such a randomized model. $\hat{f}$ is now a function of Y. Further, $\hat{f}(Y) = f(Y)$ when $Y \in \Omega_D$ by definition. Thus $\hat{f}$ is an interpolation of f. As each parameter in the model can be embedded independently, the interpolated value can be computed using a single simulation of the randomized model. Thus, the interpolation is achieved by averaging over time, instead of space.

ANALYSIS

In this section we present a general randomization scheme and prove that it produces continuous interpolations of the long-run average objective. Consider the i^{th} parameter in the model which can take integer values from the set D_i. Let $D_i = \{x_i^1, x_i^2, \ldots, x_i^p\}$ and let $C_i \subset \mathbb{R}$ denote the range $[x_i^1, x_i^p]$. To embed the i^{th} parameter into a continuous space and assign to it some value $y_i \in C_i$, we perturb the parameter periodically and assign to it values of a discrete random variable $\gamma_i(y_i)$. In general, the random variable γ_i can have a multi-point distribution, taking values from the set D_i. Let $\alpha_i^1, \alpha_i^2, \ldots, \alpha_i^p$ be a set of functions which map values from the domain C_i to the range $[0, 1]$ such that:

- $\sum_{k=1}^{p} \alpha_i^k(y) = 1 \quad \forall y \in C_i,$ (2)
- $\alpha_i^k(y) = 1$ when $y = x_i^k$ for $k \in \{1, 2, \ldots, p\},$ (3)
- $\alpha_i^1, \ldots, \alpha_i^p$ are continuous at all points in C_i. (4)

The random variable $\gamma_i(y)$ then has the distribution:

$$\gamma_i(y) = x_i^k \text{ with prob } \alpha_i^k(y) \quad \text{for } k \in \{1, 2, \ldots, p\}. \quad (5)$$

There are an infinite number of choices for the set of functions $\{\alpha_i^1, \ldots, \alpha_i^p\}$ that satisfy conditions (2) to (4). We illustrate one example of such a set in Figure 1. (In general, the shape of these functions will affect the resulting interpolation $\hat{f}$.) Let $X^1, X^2, \ldots, X^m$ be points in the n-dimensional discrete parameter space Ω_D. At each parameter point X^j we assume that the behavior of the system can be described as a stationary Markov chain with a finite state-space $\mathcal{S}$ and a transition probability matrix P^j. Such a chain will have a unique stationary distribution (that is, a unique value of the distribution π which satisfies $\pi P^j = \pi$) unless it contains two or more closed communicating classes. We assume that at each point $X^j \in \Omega_D$ the corresponding chain contains exactly one closed communicating class and therefore

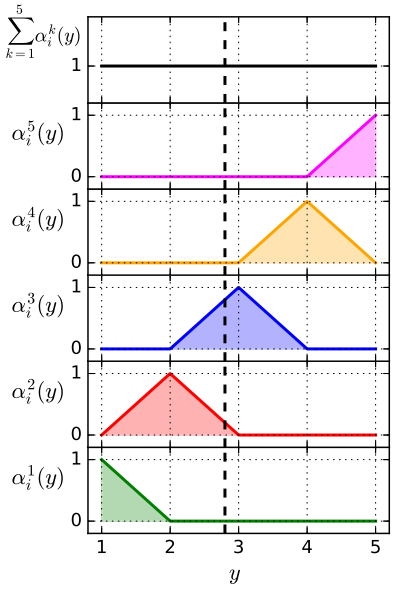


Figure 1: An example for the set of functions $\alpha_i^1, \alpha_i^2, \ldots$ that satisfy conditions (2)-(4) over the domain $D_i = \{1, 2, \ldots, 5\}$. At a given point $y = y_i$, $\alpha_i^k(y_i)$ represents the probability with which the random variable γ_i takes the value k. For example, at $y = 2.8$ indicated by the dashed line, $\alpha_i^2 = 0.2$ and $\alpha_i^3 = 0.8$ whereas $\alpha_i^k = 0$ for $k \notin \{2, 3\}$.

has a unique stationary distribution π^j. The chain is not required to be ergodic and may contain periodic states and/or some transient states. Further, we assume that the chains at $X^1, \ldots, X^m$ all share a common state-space $\mathcal{S}$. Note that it is permissible for the subset of states forming a closed communicating class at points X^i and X^j to be different or altogether non-overlapping for $i \neq j$.

Let $c : \mathcal{S} \to \mathbb{R}$ be a cost function that assigns a fixed cost to every occurrence of a state in the Markov chain. Let π_s^j denote the probability of occurrence of a state $s \in \mathcal{S}$ under the distribution π^j. The long-run average cost f at the point X^j can then be defined in terms of the stationary distribution as follows:

$$f(X^j) = \sum_{s \in \mathcal{S}} \pi_s^j c(s). \quad (6)$$

Now consider the randomized model at $Y = (y_1, y_2, \ldots, y_n) \in \Omega_C$ where the i^{th} parameter in the model is assigned values of the random variable $\gamma_i(y_i)$. Here $\gamma_1, \gamma_2, \ldots, \gamma_n$ are independent random variables with the distribution given by Equation (5). We assume that all parameters in the model are perturbed at identical time instants. Let the set of functions $\{\alpha_i^k : C_i \to [0, 1] \mid i \in \{1, 2, \ldots, n\}, k \in \{1, 2, \ldots, |D_i|\}\}$ used during randomization, be continuous and have K continuous derivates, where K is some non-negative integer (that is, let each function be of differentiability class C^K where $K \in \{0, 1, 2, \ldots\}$.) Let $\hat{f} : \Omega_C \to \mathbb{R}$ be the corresponding long-run average measure of the randomized model. Then, we show that the following statement is true:

Theorem 1. *$\hat{f}$ is a continuous interpolation of f and also belongs to differentiability class C^K.*

Proof. The n-dimensional vector of parameter values at any instant of time is itself a random variable Γ whose distribution is a function of Y as follows:

$$\Gamma(Y) = X^j \text{ with prob } \beta^j(Y) \quad \text{for } j \in \{1, 2, \ldots, m\}. \tag{7}$$

where m is the total number of points in the discrete design space ($m = \prod_{i=1}^{n} |D_i|$). Let $X^j = (x_1^j, x_2^j, \ldots, x_n^j)$ be a point in Ω_D and let $I_i(x)$ denote the index of element x in the set D_i. The coefficients β^j can then be obtained as:

$$\beta^j(Y) = \prod_{i=1}^{n} \mathbb{P}(\gamma_i(y_i) = x_i^j) \quad = \prod_{i=1}^{n} \alpha_i^{I_i(x_i^j)}(y_i). \tag{8}$$

From equations (8) and (4) it follows that the functions $\beta^1, \beta^2, \ldots, \beta^m$ which map values from the domain Ω_C to the range $[0, 1]$ also satisfy the following conditions:

- $\sum_{j=1}^{m} \beta^j(Y) = 1 \quad \forall Y \in \Omega_C,$ (9)
- $\beta^j(Y) = 1$ when $Y = X^j$ for $j \in \{1, 2, \ldots, m\}$, (10)
- $\beta^1, \ldots, \beta^m$ are continuous at all points in Ω_C. (11)

Let $P(Y)$ denote the transition probability matrix of the randomized model. We choose the time instants at which to perturb the parameter values in such a way that $P(Y)$ is given by

$$P(Y) = \sum_{j=1}^{m} \beta^j(Y) P^j. \tag{12}$$

In a discrete-time system, this can be achieved in a straightforward manner by perturbing the parameter values at the beginning of each time-slot. From (12) and (11) it follows that $P(Y)$ is continuous with respect to Y. We now refer to a result from (Schweitzer 1968; Section 6) which states that: *If P^A is the transition probability matrix of a finite Markov chain containing a single irreducible subset of states (a single closed communicating class), then for an arbitrary stochastic matrix P^B with the same state-space as P^A, the randomized stationary Markov chain with transition probability matrix*

$$P(\lambda) = (1 - \lambda)P^A + \lambda P^B \quad 0 \leq \lambda < 1$$

will also possess a single irreducible subset of states. Further, $P(\lambda)$ has a unique stationary distribution $\pi(\lambda)$ which is infinitely differentiable with respect to λ for $\lambda \in [0, 1)$.

In Equation (12) $P^1, P^2, \ldots, P^m$ each have a single irreducible set of states. Therefore the stationary distribution $\pi(Y)$ corresponding to $P(Y)$ exists and is infinitely differentiable with respect to the coefficients $\beta^1(Y), \ldots, \beta^m(Y)$ and K-times continuously differentiable (C^K) with respect to Y. Let $\pi_s(Y)$ denote the probability of occurrence of a state s under the distribution $\pi(Y)$. The long-run average cost $\hat{f}$ in the randomized model is given by:

$$\hat{f}(Y) = \sum_{s \in \mathcal{S}} \pi_s(Y) c(s)$$

The function $\hat{f}$ is also C^K with respect to Y. From equations (7) and (10) we have $\Gamma(Y) = X^j$ with probability 1 when $Y = X^j$ for $j \in \{1, 2, \ldots, m\}$. Thus $\hat{f}(Y) = f(Y)$ whenever $Y \in \Omega_D$. Therefore $\hat{f}$ is a C^K interpolation of f. □

Thus we have shown that a randomization of the simulation model can be used as a means of producing continuous interpolations of the long-run average measure f under the listed assumptions. It should be noted that for $\hat{f}$ to be of class C^K, it is sufficient but not necessary for the coefficient functions $\alpha_i^1, \alpha_i^2, \ldots$ to be C^K functions. For instance, it may be possible to obtain a continuously differentiable interpolation $\hat{f}$ using coefficient functions that are continuous but not differentiable at the integer points.

APPLICATION TO DISCRETE-TIME QUEUES

In this section we describe the application of the embedding technique to discrete-time queues for embedding queue-capacity, number of servers and service-time parameters into a continuous space and present simulation results for several concrete examples.

We assume that in each slot, jobs arrive into the system near the beginning of a slot and depart towards the end of the slot. At-most one job can arrive within a single slot. S_0 denotes the initial state of the queue and S_t denotes the state measured at the end of slot t. For job arrivals with geometrically distributed inter-arrival times (denoted Geo) the probability of a job arriving in a slot is denoted as p. For geometrically distributed (Geo) service times, the probability of the server finishing an ongoing job in the current slot (irrespective of the amount of time for which the job has already received service) is denoted as q. For a deterministic server (denoted D), every job takes a constant amount of time to be processed by the server. The number of slots taken to process a job is denoted as T.

For performing the randomization, we will use a more general form for the random variable γ, which was first introduced in Equation (1) as follows:

$$\gamma(y) = \begin{cases} \lfloor y \rfloor & \text{with probability } \alpha^1(y) = \alpha(y) \\ \lceil y \rceil & \text{with probability } \alpha^2(y) = 1 - \alpha(y) \end{cases}$$

$$\text{where } \alpha(y) = \begin{cases} \dfrac{\lceil y \rceil^s - y^s}{\lceil y \rceil^s - \lfloor y \rfloor^s} & \text{when } y \notin \mathbb{Z} \\ 0 & \text{when } y \in \mathbb{Z} \end{cases} \tag{13}$$

where $s \in \mathbb{R}$ is a constant. It can be seen that the probability functions $\alpha^1(y)$ and $\alpha^2(y)$ will satisfy the conditions (2)-(4) for $s \neq 0$.

Embedding Queue-Capacity

Consider a Geo/Geo/1 queue with finite buffering. The queue state S_t is the number of jobs in the queue at the end of slot t. The queue has a capacity parameter $C \in \mathbb{N}$ that we wish to embed into a continuous space. We first define the behavior of the queue with respect to C as follows:

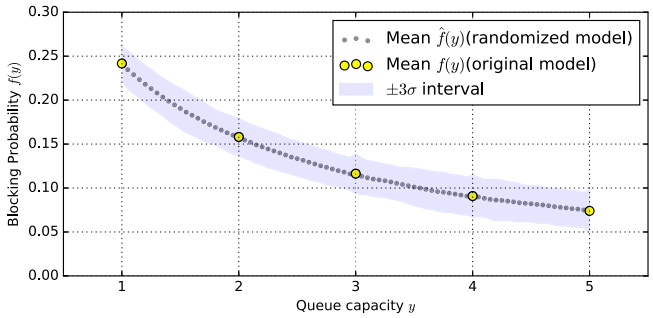


Figure 2: Interpolation produced using the randomization-based embedding technique for a finite-capacity Geo/Geo/1 queue. The embedded parameter y is the queue-capacity and $\hat{f}(y)$ is the blocking probability of the queue. (Simulation length=10^4 slots, $p = 0.5, q = 0.51$).

Definition 1. *A job arriving in slot t is allowed to enter the queue if $S_{t-1} < C$, else the job is lost.*

By defining the capacity parameter in this way, we ensure that the Markov chains corresponding to every possible value of C share the same state-space. For a queue with a constant value of C, the states $\{S_t \mid S_t > C\}$ are transient and unreachable from states $\{S_t \mid S_t \leq C\}$. The definition also makes it intuitive for the parameter C to be updated dynamically within a simulation, in the randomized model. Let $f : \mathbb{N} \to \mathbb{R}$ be some long-run average measure of this system expressed as a function of the capacity parameter C. We wish to embed the capacity parameter into continuous space, and evaluate the interpolation $\hat{f}(y)$ of f at some given point $y \in \mathbb{R}_{\geq 1}$. To do so, we construct a randomized model where the capacity parameter is perturbed at the beginning of each slot and assigned the value of the random variable $\gamma(y)$ as described in Equation (13) with $s = -1$. Let C_t denote the instantaneous value of the capacity parameter for the duration of slot t. By the definition of the capacity parameter above, whenever the parameter is updated the jobs already present in the queue are not disturbed. The updated value of the parameter is used solely to decide if a new job should be accepted into the queue. Thus it is possible that $S_t > C_t$ at some time instants t.

In Figure 2, we show the simulation results obtained with this randomization scheme. We fix the arrival and service probabilities p, q and sweep the queue capacity parameter y in steps of 0.05. The interpolated function $\hat{f}(y)$ is the blocking probability (probability of an arriving job being denied entry into the system). Each point in the plot is the mean value of $\hat{f}(y)$ measured using 100 simulation samples with distinct randomization seeds. The shaded area around the plot represents the ± 3 standard-deviation interval.

We observe that a randomization of the model produces a smooth interpolation of the objective. The standard deviation values (which represent the simulation error) are similar at the discrete and the interpolated points, indicating that the stochastic error contributed by the randomization is negligible. The computational overhead of the embedding, contributed primarily by the additional calls to a random number generator, was found to be between 5% to 10%. The overhead was computed as the relative difference between the time per simulation for the original discrete-parameter model (at some $y = y_0 \in \mathbb{N}$) and the randomized model (at $y_0 + 0.5$). This overhead is very small in comparison to the computational cost involved when using spatial interpolation.

Thus a smooth embedding of the queue capacity parameter could be obtained efficiently through a randomization of the simulation model.

The shape of the interpolation curve is sensitive to the choice of the interpolation coefficients $\alpha^1(y), \alpha^2(y)$, and thus the value of the parameter s used for generating these coefficients. For most of the examples considered in this study, we observe that setting $s = 1$ results in a smooth interpolation. For other examples, we have tuned s to obtain a smooth interpolation. It may be possible to arrive at the best randomization settings (the choice of the functions α^k) analytically rather than through tuning. However this is beyond the scope of the current work and can be an interesting direction for future study.

Embedding the Number of Servers

Consider a Geo/Geo/K queue. The queue has infinite buffering and K identical, independent servers working in parallel. We wish to embed the parameter $K \in \mathbb{N}$ into continuous space. To do so, we first re-define the system behavior as follows:

Definition 2. *The system consists of a single, infinitely fast* controller *with a parameter K, and a fixed large number ($> K$) of identical, independent servers. In each time-slot, the controller pulls jobs from the head of the queue and assigns each job to a free server, as long as the queue is not empty and the number of jobs currently receiving service is less than K.*

Note that for a fixed (integer) value of K, this description is identical to a queue with K independent servers. However, the new definition of the queue behavior makes it intuitive for the controller parameter K to be updated dynamically. To embed K into continuous space and evaluate the interpolation at some point $y \in \mathbb{R}_{\geq 1}$, we construct a randomized model where K is perturbed at the beginning of each slot and assigned the value of the random variable $\gamma(y)$ defined in Equation(13) with $s = 1$. By the definition above, whenever the parameter K is updated, the jobs already receiving service are not disturbed and the updated parameter value is used solely for deciding if service can commence for new jobs.

In Figure 3, we show the interpolation obtained using this scheme. We fix the arrival and service probabilities p, q and sweep the parameter y in small steps (the step-size is chosen to be smaller near the knee region). The interpolated function $\hat{f}(y)$ is the average number of jobs in the system. Each point in the plot is the mean value of $\hat{f}(y)$ measured using 100 simulation samples. We observe that a randomization of the

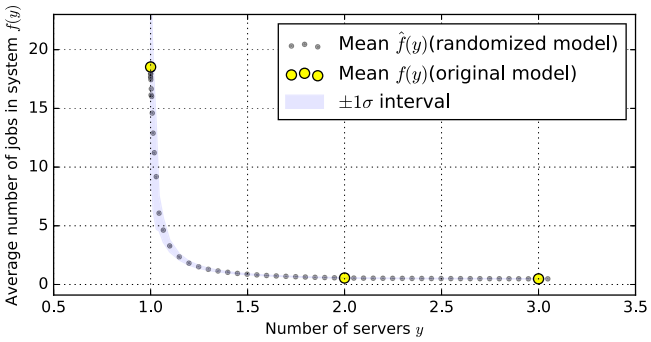


Figure 3: Interpolation results for embedding the number of servers (K) in a Geo/Geo/1/K queue. The embedded parameter $y = K$ is the number of servers and the interpolated function $\hat{f}(y)$ is the average number of jobs in the system. ($p = 0.5$, $q = 0.51$, simulation length$= 10^4$ slots)

model produces smooth interpolations of f in a Geo/Geo/1/K queue. Similarly, for a queue with a deterministic service time (a Geo/D/1/K queue), a smooth interpolation was obtained using the same randomization settings.

Embedding Service Time

Consider a Geo/D/1 queue. The server is deterministic with a fixed service time of $T \in \mathbb{N}$ slots. To embed the parameter T into continuous space, we first define the server behavior as follows:

Definition 3. *The server has a parameter T. At the end of each slot, the server ends jobs that have already received $\geq T$ slots of service.*

To embed T into continuous space and evaluate the interpolation at some point $y \in \mathbb{R}_{\geq 1}$, we construct a randomized model where T is perturbed at the beginning of each slot and assigned the value of the random variable $\gamma(y)$ defined in Equation (13) with $s = 1$. In Figure 4, we show the interpolation obtained using this randomization scheme. We fix the arrival probability p and sweep the service time parameter y in steps of 0.05. The interpolated function $\hat{f}(y)$ is the average number of jobs in the system. Each point in the plot is the mean value of $\hat{f}(y)$ measured using 100 simulation samples. The randomization produces a smooth interpolation of f.

Using the approach described in this section, multiple discrete parameters in a model can be embedded simultaneously and independently of each other, and existing continuous optimizers can be applied over such an embedding.

AN OPTIMIZATION CASE STUDY

To demonstrate the utility of the embedding technique, we present an optimization case study for a queueing network shown in Figure 5. We optimize this queueing network using the embedding-based approach described in this paper and compare its performance to a direct discrete-space search method. Although the focus of this paper is on the embedding technique itself (and not on a comparison between optimization algorithms), the preliminary results obtained for this case-study indicate that descent-based continuous optimizers can be effectively applied over the embedding and may perform significantly better in comparison to discrete-space methods.

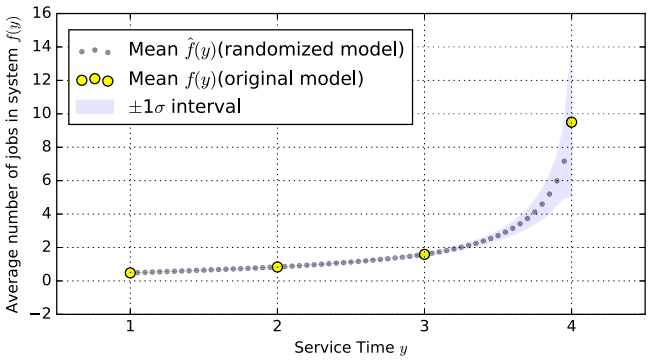


Figure 4: Interpolation results obtained by embedding the service-time in a Geo/D/1 queue. The embedded parameter y is the service time (in slots) and $\hat{f}(y)$ is the average number of jobs in the system. ($p = 0.24$, simulation length$= 10^4$ slots)

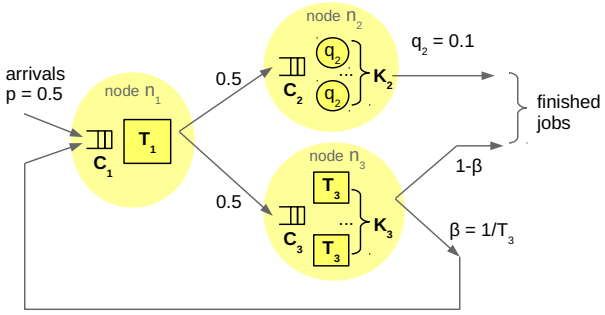
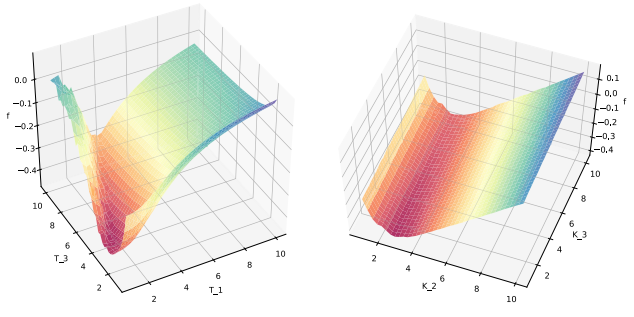


Figure 5: Queueing network to be optimized

Problem Statement: The queueing network in Figure 5 consists of three nodes n_1, n_2 and n_3. Each node n_i has a queue with a finite capacity C_i in front of it. Jobs arrive at n_1 with geometrically distributed inter-arrival times (with an arrival probability p). An arriving job that finds the queue full is lost. The node n_1 consists of a single deterministic server with a service time of T_1 slots. This server forwards each job to either n_2 or n_3 with equal probabilities, and stalls if the destination queues are full. Nodes n_2 and n_3 respectively consist of K_2 and K_3 identical servers working in parallel. The servers in n_2 have geometrically distributed service times (with service probability q_2) whereas those in n_3 are deterministic, with a service time of T_3 slots. The servers in T_3 are prone to faults. The probability of a job turning out faulty (denoted β) is inversely related to the service time ($\beta = 1/T_3$). A job that has received faulty service is sent back to node n_1 to be re-processed as a fresh job. If the destination queue at n_1 is full, the corresponding server in n_3 stalls. The arrival probability p and the service parameter q_2 are kept fixed ($p = 0.5$, $q_2 = 0.1$). The parameter set for this system is $\{C_1, C_2, C_3, T_1, T_3, K_2, K_3\}$. Each parameter can take integer values between 1 and 10. Thus the parameter



(a) $f(T1, T3)$ (b) $f(K2, K3)$

Figure 6: The interpolated objective function $\hat{f}$ (obtained via simulation of the queueing network in Figure 5) plotted along 2-dimensional slices of the parameter space.

space has 10^7 design points.

Objective Function: Let X denote the vector of parameter values and Ω_D denote the set of all possible values that X can take. We define the objective function $f(X)$ as a weighted sum of throughput and cost components. The throughput component, denoted $T(X)$ is the expected value of the long-run average throughput of the system (estimated using simulation). The cost component, denoted $C(X)$ is modeled using a synthetic function as follows:

$$C(X) = (C_1 + C_2 + C_3) + \frac{20}{T_1} + 100K_2 + 20\frac{K_3}{T_3}. \quad (14)$$

Thus the cost increases with increasing buffer sizes and the number of servers, and reduces with increasing service-times. The objective function to be minimized is a weighted sum of the normalized cost and throughput components, defined as follows:

$$f(X) = \frac{C(X)}{\max\limits_{j \in \Omega_D} C(j)} - \frac{T(X)}{p} \quad (15)$$

Since an exhaustive evaluation of f over all 10^7 design points is infeasible, our goal is to find the best solution possible within a fixed computational budget. To solve this optimization problem, we first embed the discrete parameter space into a continuous one by randomizing each parameter in the model using the technique described in the previous section. To understand the nature of the resulting interpolation (in terms of continuity, convexity and smoothness), we plot the interpolated objective along arbitrary 2-dimensional slices of the 7-dimensional domain. Figure 6 shows the interpolated objective $\hat{f}$ plotted along two such slices. The plots were obtained by sweeping two parameters at a time (in steps of 0.25) while keeping the other parameter values fixed. Each point on this plot is obtained via a single simulation of the randomized model of length 10^4 slots. Along each slice, the interpolation is found to be reasonably smooth. The plots obtained along several other slices in the domain were similar, indicating that the interpolation obtained via the randomization is well-suited to the application of continuous-space optimizers.

We evaluate the performance of two continuous optimizers, COBYLA (Powell (1994; 2003)) and SPSA (Spall (1992)) applied *directly* over the randomization-based embedding and compare their performance against a discrete-space version of SPSA (Whitney et al. (2001)) applied over the original discrete-parameter model. While discrete-parameter variants of SPSA exist, COBYLA has not been conventionally applied in the discrete-parameter case. This fact illustrates the utility of our embedding technique, which makes it possible for a large set of existing continuous optimizers to be applied for solving discrete-parameter problems. Both COBYLA and SPSA were chosen as they do not require an explicit computation of numerical derivatives along each parameter axis and are thus suited to high dimensional problems. Further, both methods are suited to problems where the objective function evaluations can be noisy. The settings for each optimizer were selected via tuning and were identical across the continuous and discrete versions of SPSA.

Results: For each optimization method, we perform 100 optimization runs using distinct, randomly chosen initial points. The set of initial points is fixed and is common across all three optimization methods. For each optimization run we set an upper limit of 1000 objective evaluations. At each objective function evaluation, the system throughput is measured using a single simulation of the randomized model (of length 10^4 slots) and the cost is computed analytically using Equation (14). At the end of each optimization run, we round the solution to the nearest integer point, and record the objective value at this point. Table 1 presents the performance results for the three methods measured across 100 optimization runs. The results show that COBYLA shows the best performance, both in terms of the quality of the solutions and the convergence rate. The continuous-space SPSA performs better in comparison to its discrete-parameter variant. The solutions were found to be well-clustered. (Among the top 20 solutions found by COBYLA, all were found to have the parameter values $T_1 = 1$, $T_3 = 10$, and $K_2 = 3$.) The results indicate that existing continuous optimizers can be effectively applied over the embedding and their performance compares favourably against direct discrete-space search.

		COBYLA	**SPSA**	**Discrete-SPSA**
Objective value at the optimum (lower is better)	best	-0.7130	-0.7108	-0.6842
	avg	-0.5240	-0.1994	-0.1964
	std-dev	0.2930	0.3481	0.3358
Avg number of objective function evaluations per optimization run		52.7	1000	1000
Avg time per optimization run (seconds)		0.15	2.94	2.33

Table 1: Performance of two continuous optimizers (COBYLA and SPSA) applied directly over the randomization-based embedding, and a discrete-space optimizer (Discrete-SPSA) applied over the original discrete-parameter space.

CONCLUSIONS

This paper presented a simple and computationally efficient technique using which discrete parameters in queueing systems can be embedded into a continuous-space, enabling direct application of continuous-space methods for simulation-based optimization. The technique is based on a randomization of the parameter values in the simulation model and is applicable to problems where the objective function is a long-run average measure. Unlike spatial interpolation, the interpolated value can be measured using a single simulation of this randomized model irrespective of the dimensionality of the design space. We presented a theoretical basis for this embedding technique and described its application to discrete-time queues for embedding queue capacities, number of servers and service-time parameters into a continuous space. We then demonstrated the utility of this embedding technique via an optimization case study of a queueing network with 7 parameters. A randomization of the simulation model produced reasonably smooth and continuous interpolations of the objective. Two continuous-space optimizers (COBYLA, SPSA) applied directly over this embedding were found to perform better in comparison to a direct discrete-space search (Discrete-SPSA).

The shape of the generated interpolation curves is affected by the choice of the randomization settings (in particular, the choice of the functions $\alpha_i^1, \alpha_i^2, \ldots$ used during randomization). For obtaining a smooth interpolation, we have performed a tuning of these randomization parameters in the current work. However, for some systems it may be possible to arrive at the best randomization scheme analytically, and this can be a direction for future investigation. Further, the effect of the perturbation interval on the generated interpolation curves also needs to be investigated further. A limitation of this embedding technique is that it may require a modification of the simulation program. Some state transitions that were not possible in the original (fixed-parameter) model may now become possible when the parameter values are randomized. These transitions have to be accounted-for by the programmer.

In summary, this paper showed that the randomization-based embedding technique can be a useful and effective tool in simulation-based optimization of queueing systems. Future work can extend the embedding technique to other kinds of discrete-event systems such as inventory models.

REFERENCES

Alfa A.S., 2015. *"Applied Discrete-Time Queues"*. Springer Publishing Company, Incorporated, 2nd ed. ISBN 149393418X, 9781493934188.

Bhatnagar S. and Kowshik H.J., 2005. "A Discrete Parameter Stochastic Approximation Algorithm for Simulation Optimization". *SIMULATION*, 81, no. 11, 757–772.

Bhatnagar S.; Mishra V.K.; and Hemachandra N., 2011. "Stochastic Algorithms for Discrete Parameter Simulation Optimization". *IEEE Transactions on Automation Science and Engineering*, 8, no. 4, 780–793. ISSN 1545-5955.

Chen C.H. and Lee L.H., 2010. *"Stochastic Simulation Optimization: An Optimal Computing Budget Allocation"*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1st ed. ISBN 9789814282642, 9814282642.

Karanjkar N.V. and Desai M.P., 2015. "An Approach to Discrete Parameter Design Space Exploration of Multicore Systems Using a Novel Simulation Based Interpolation Technique". In *Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), 2015 IEEE 23rd International Symposium on*. ISSN 1526-7539, 85–88.

Lim E., 2012. "Stochastic Approximation over Multidimensional Discrete Sets with Applications to Inventory Systems and Admission Control of Queueing Networks". *ACM Trans Model Comput Simul*, 22, no. 4, 19:1–19:23. ISSN 1049-3301.

Powell M., 2003. "On Trust Region Methods for Unconstrained Minimization without Derivatives". *Mathematical Programming*, 97, no. 3. ISSN 0025-5610.

Powell M.J.D., 1994. *"A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation"*, Springer Netherlands, Dordrecht, chap. 4. ISBN 978-94-015-8330-5, 51–67.

Schweitzer P.J., 1968. "Perturbation Theory and Finite Markov Chains". *Journal of Applied Probability*, 5, no. 2, 401–413. ISSN 00219002.

Spall J.C., 1992. "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation". *IEEE Transactions on Automatic Control*, 37, no. 3, 332–341.

Swisher J.R.; Hyden P.D.; Jacobson S.H.; and Schruben L.W., 2000. "A Survey of Simulation Optimization Techniques and Procedures". In *Simulation Conference, 2000. Proceedings. Winter*. vol. 1, 119–128 vol.1.

Wang H. and Schmeiser B.W., 2008. "Discrete Stochastic Optimization using Linear Interpolation". In *2008 Winter Simulation Conference*. IEEE. ISBN 978-1-4244-2707-9, 502–508.

Whitney J.E.; Solomon L.I.; and Hill S.D., 2001. "Constrained optimization over discrete sets via SPSA with application to non-separable resource allocation". In *Proceeding of the 2001 Winter Simulation Conference (Cat. No.01CH37304)*. vol. 1, 313–317 vol.1.

WEB REFERENCES

Karanjkar N., 2017. *Online repository of queueing models and scripts used for the embedding experiments*. `https://github.com/NehaKaranjkar/Embedding`.

MACHINE LEARNING AND AI

APPLICATION OF MACHINE LEARNING TO MODEL A BIOLOGICAL REACTOR IN A WASTEWATER TREATMENT PLANT

Bartosz Szeląg[*], Jan Studziński[**]
[*] Kielce University of Technology, 25-314 Kielce, Poland
bszelag@tu.kielce.pl
[**] Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw, Poland
studzins@ibspan.waw.pl

KEYWORDS
Wastewater treatment plant, activated sludge concentration models, MLP, SVM, MARS methods.

ABSTRACT

The paper presents the results of modelling the concentration of activated sludge in a biological reactor of a flowing sewage treatment plant. The value of this parameter is an important indicator of the condition of the plant operation and its forecasting with the use of a mathematical model enables to make the right operational decisions for the plant technologist. Three methods of machine learning were used for modelling: neural networks, the method of support vectors SVM and the MARS method. In the first stage of investigation, the sediment concentration was modelled using direct measurements of the raw wastewater inflow and the wastewater quality at the inflow to and outflow from the plant, while in the second stage, the time-consuming measurements of BOD_5 have been replaced by its model based on the wastewater inflow. The best modelling results were obtained in both cases using the MARS method. The obtained results of calculations using an indirect modelling method are satisfactory and at the same time such a hybrid model can be used in the plant continuously to forecast the reactor operating state, which is not possible in the case of a model using BOD_5 measurements directly.

INTRODUCTION

Wastewater treatment is a complex process and requires a range of operational parameters to be maintained. This is a key task in order to obtain the values of selected sewage quality indicators at the outlet of the facility below the limit values set by the relevant legal regulations. The operation of a biological reactor is influenced by a number of operational parameters, which include in particular the concentration of activated sludge, depending on the degree of sludge recirculation, the amount of chemical coagulants dosed, the concentration of dissolved oxygen, etc. In most cases the values of these parameters are not set optimally, so in many cases, both in summer and winter, the selected quality indicators of wastewater discharged from the treatment plant are exceeded. On the other hand, we can observe the operation of facilities where the values of appropriate settings are so high that the values of indicators at the outflow from the treatment plant are below the limit values, but the process being carried out is uneconomical and generates high operating costs. Therefore, in order to improve the efficiency of sewage treatment plants and at the same time reduce their operating costs, it is necessary to support the treatment process with the use of mathematical models. Physical models based on systems of differential equations describing kinetics of biochemical reactions may be used for modelling of treatment plants, however, due to the large number of parameters they contain, problems with their calibration usually occur (Henze et al. 1999, Szetela and Dymaczewski 2002). Therefore, statistical models (grey and black box) are also used to simulate a biological reactor, in which both the structure and values of the model parameters are determined on the basis of measurement data collected at the plant during the operation of the object.
One of the most important operating parameters of a sewage treatment plant is the concentration of activated sludge, as its value has a key impact on the substrate load of the bioreactor and the age of the sludge, which in turn determine the quality of sewage and the course of processes in the activated sludge chambers. Currently, most of the modelling work on biological reactors using statistical models uses only measured data on wastewater quality at the plant inlet (Szeląg et al. 2018, Guclu and Dursun 2010). This approach is questionable from the point of view of process physics (activated sludge treatment of wastewater) and may lead to erroneous technological decisions affecting the operation of the biological reactor and the efficiency of the sewage treatment plant. Therefore, the present paper presents a model for forecasting the concentration of activated sludge, taking into account the quantity and quality of sewage at the inflow to the plant, as well as the bioreactor operation parameters and values of selected indicators of sewage quality at the outflow from the sewage treatment plant. For simulation studies the methods of neural networks (Multilayer Perceptron MLP), Support Vector Machines (SVM) and Multivariate Adaptive Regression of Spline (MARS) were used.

THE OBJECT OF INVESTIGATION

The modelling and simulation calculations were based on data from a 72.000 m^3/d municipal wastewater treatment plant with a Population Equivalent (PE) load capacity of 275.000. The plant receives sewage from the city of Kielce and two neighbouring municipalities. Wastewater treatment in the facility is carried out first mechanically (by stepwise grids and aerated sandstone with separated grease removal) and then biologically on the basis of a modified Bardenpho system with a pre-denitrification chamber. The treated wastewater flows to a secondary clarifier, from where it is

separated from the active sludge and discharged to a reservoir - the Bobrza river.
Within the monitoring system carried out at the sewage treatment plant, the quality of sewage is measured (biological and chemical oxygen demand, ammonium nitrogen, total nitrogen, suspended matter, total phosphorus) no less than 6 times a month at the inflow and outflow of the facility from 2012 onwards. Simultaneously, on-line measurements are carried out of the amount of sewage flowing into the treatment plant and the parameters of the biological reactor (pH, sediment temperature, oxygen concentration, active sediment concentration, degree of recirculation, amount of excessive sediments discharged, amount of PIX coagulant dosed).

METHODOLOGY

In the calculations performed, a mathematical model was developed to simulate the active sediment concentration values using the methods of MLP, SVM and MARS. A model for predicting the concentration of the sediment was proposed in the following form (Szeląg and Gawdzik 2017):

$$MLSS = f(Q, C_{in,eff}, T_{AS}, m_{met}, m_{PIX}, DO, RAS, SE) \quad (1)$$

where Q - the amount of sewage inflow, T_{AS} - the temperature in the activated sludge chambers, m_{met} - the daily dose of methanol, m_{PIX} - the daily dose of PIX coagulant, RAS - the degree of sludge recirculation, SE - the sludge settle-ability indicator, DO - the concentration of dissolved oxygen, $C_{in,eff}$ - the values of selected sewage quality indicators at the inlet (inflow) and outlet (effluent) of the treatment plant, including: BOD_5 - biochemical oxygen demand, COD - chemical oxygen demand, TSS - general suspension, TN - total nitrogen, N-NH4 - ammonium nitrogen, TP - total phosphorus.
Due to the time-consuming determination of BOD_5, it was assumed in the analyses that its value would be modelled on the basis of the relation:

$$BOD_{5,in} = f(Q(t-1), Q(t-2), .., Q(t-i)) \quad (2)$$

where Q(t-i) are the sewage inflow values to the treatment plant preceding the projected BOD_5 at time t.
As a result, the MLSS prediction model described by equations (1-2) enables to simulate the important parameter of the plant operation continuously, which is a significant advantage compared to the works of other authors (Guclu and Dursun 2010, Hong and Bhamidimaria 2003).
Neural networks of multilayer perceptron type (MLP) used in the study are commonly applied to model phenomena where the relation between inputs (x_i) and output (y) is non-linear (Dogan et al. 2008, Ossowski 2013). In such a model, first I - input data (x) are multiplied by the values of the weights (w) and transmitted to J - hidden layer neurons, in which their summation takes place, and then the obtained sums are transformed using the linear or non-linear activation functions f(-) and transferred to the output neurons. The values of weights in the model are estimated at the stage of network learning by means of appropriate numerical algorithms. The values of network outputs (y_m) are calculated from the formula:

$$y = \sum_{j=1}^{J} w_j f\left(\sum_{i=1}^{I} w_{ij} \cdot x_i + b_j\right) \quad (3)$$

where: I - number of model inputs, J - number of neurons in the hidden layer, w_{ij} - values of weights between the inputs and neurons of the hidden layer, b_j - biases of neurons of the hidden layer, w_j - values of weights between the neurons of the hidden layer and the output from the model, f(-) - activation function.
In the calculations at the stage of creation of the MLP model, in order to limit potential model overlearning, the number of neurons in the hidden layer was assumed from the range of <j÷2j+1> (where j - the number of model inputs) and the following activation functions in the hidden and output layers were considered: linear, exponential, hyperbolic tangent, logistic, linear and sinusoidal. The BFGS (Broyden-Fletcher-Goldfarb-Shanno) method was used to determine the weights of the MLP model.
A modification of the method of artificial neural networks of the multi-layer perceptron type is the SVM method of support vectors, in which the neurons in the hidden layer of the network were replaced by nuclear functions (kernel) (Burges 1998). The application of nuclear functions in the model allows for the transformation of non-linear N-dimensional space into a K-dimensional linear space of characteristics with a larger dimension. In the calculations, a regressive method of support vectors with radial nuclear function was applied in order to minimize the functional of the form:

$$\sum_{i=1}^{m} \frac{c}{m} |y_i - y(x_i)|_{\varepsilon} + \frac{1}{2} \cdot \|f\|_k^2 \quad (4)$$

where: $|y_i\text{-}y(x_i)|_\varepsilon = \max\{0, |y_i\text{-}y(x_i)\text{-}\varepsilon|\}$, ε - permissible error rate, ||f| - norm f in the Hilbert space, c - user-selected constant, depending on the value of ε (Ossowski 2013), m - quantity of the teaching set, $y(x_i)$ - function value f=y(x) in point x_i described by the equation:

$$y = \sum_{i=1}^{N_{sv}} (\alpha_i - \alpha_i^{'}) K(x, x_i) + w_0 \quad (5)$$

where: w_0 - deviation, Nsv - number of support vectors dependent on C and ε, α_i, α_i' - Lagrange multipliers, $K(x,x_i)$ - nuclear function with radial base functions (Ossowski 2013).
In models developed using MLP and SVM methods, optimal values of parameters describing the structure of models were searched while using the method of successive approximations, substituting successive values of parameters until the moment of obtaining appropriate values of adjustment measures, such as correlation coefficient (R), mean absolute error (MAE), mean relative error (MAPE) and mean square error (RMSE).
The MARS method is used for solving regression problems (Friedman 1991, Szeląg i Studziński 2017). In the classical regression approach, independent variables are treated uniformly, whereas in the MARS method variation ranges of the concerned predictors are divided into subranges in which

independent variables can have different impacts on the phenomenon investigated. The boundaries of subranges are determined on the basis of threshold values (t). The values of the predictors are segregated into those lower and higher than threshold values t_i. That is performed by means of the basic functions that have the following form:

$$h(X) = \alpha_i (\max(0, X - t) \quad (6)$$

where h(X) – vector of the basic functions for individual variables (x_i) for which the following condition is satisfied:

$$x_i - t_i = \begin{cases} x_i - t_i & for \quad x_i > t_i \\ 0 & for \quad x_i \le t_i \end{cases} \quad (7)$$

In the MARS method the regression relation is a spline function obtained from a linear combination of the product of the basis functions and appropriate weights. The function can be written as follows:

$$y = \alpha_0 + \sum_{m=1}^{M} \alpha_m \cdot h_m(X) \quad (8)$$

where: $X=[x_1,x_2,\ldots,x_i]$ – vector of input data, α_m – weight values, h_m – basic functions.

To calculate the values of the model parameters, Friedman (1991) developed a special algorithm that allows the search of observations to establish threshold values. Software STATISTCA was used to develop the above models and the data set was divided into teaching data subset (80%) and test data subset (20%).

RESULTS

On the basis of the results of measurements of the amount of sewage at the plant inlet and the quality of sewage at the plant inlet and outlet (BOD_5, COD, TN, N-NH4, TSS, TP) as well as measurements of parameters of biological reactor operation (T_{KOC}, pH, m_{PIX}, WAS, RAS, FM, SE), the ranges of variability of these measurements (minimum, average, standard deviation) were determined; the results of the analyses are presented in Table 1. On the basis of Table 1 it can be concluded that in the analyzed time period the quantity and quality of sewage at the inflow changed to a large extent, which had a significant impact on the operational parameters of the bioreactor and the course of processes within the activated sludge chambers.

At the same time, varied quality of sewage at the facility inlet and variability of reactor parameters resulted in a change in the quality of sewage at the outlet. Based on the measurement results, models were developed to simulate the activated sludge concentration by the three methods mentioned above. Table 2 presents values of measures of adjustment of measurement results to calculations for particular methods, while Fig. 1 shows graphically comparison of calculation results with the measurements.

In the model obtained by the MLP method the number of neurons in the hidden layer was 12, and for the hidden and the input layers the hyperbolic tangent and exponential activation functions were selected, respectively. In the SVM model, the lowest error values were obtained for C=100, ε=0.10 and γ=0.70. In the model developed by the MARS method, the optimal structure of the model was obtained for 24 nodes.

Table 1. Summary of the range of variability in the amount and quality of sewage and bioreactor parameters.

Variables	Units	Minimum	Average	Standard deviation
Q	m^3/d	42.563	40.698	8.088
T_{AS}	^0C	10	15,9	3,58
pH	-	7,2	7,6	0,2
MLSS	kg/m^3	2,8	3,98	0,96
RAS	%	55,6	90,7	19,23
m_{PIX}	m^3/d	0,00	0,71	0,39
WAS	kg/d	5.489	11.123	3.950
DO	mg/dm^3	1,25	2,00	0,56
F/M	$gBOD_5/gMLSS \cdot d$	0,03	0,07	0,02
SE	ml/g	95	166	25
$BOD_{5,in}$	mgO_2/dm^3	150	309	86,01
COD_{in}	mgO_2/dm^3	405	791	173,98
TSS_{in}	mg/dm^3	136	329	77,25
$N\text{-}NH_{4,in}$	mg/dm^3	25,5	7,8	6,15
TN_{in}	mg/dm^3	32,36	77,73	10,62
TP_{in}	mg/dm^3	0,18	0,35	0,12
$BOD_{5,eff}$	mgO_2/dm^3	2,00	4,59	1,35
COD_{eff}	mgO_2/dm^3	23,00	35,40	6,35
TSS_{eff}	mg/dm^3	1,00	5,60	4,79
$N\text{-}NH_{4,eff}$	mg/dm^3	0,20	2,14	3,77
TN_{eff}	mg/dm^3	4,38	7,83	3,47
TP_{eff}	mg/dm^3	0,16	0,35	0,15

Table 2: Summary of measures to match MLSS calculation results to measurement data.

	Teaching				Testing			
	R	MAPE	MAE	RMSE	R	MAPE	MAE	RMSE
Method	-	%	kg/m^3	kg/m^3	-	%	kg/m^3	kg/m^3
MLP	0,904	8,31	0,36	0,46	0,905	8,39	0,37	0,48
SVM	0,924	7,54	0,32	0,41	0,927	7,59	0,33	0,43
MARS	0,938	6,91	0,29	0,37	0,94	6,86	0,27	0,35

The results of the MLSS modelling are slightly worse than those obtained by Hong and Bhamidimaria (2003), who based on the measurements of MLSS(t-1), RAS, pH, rainfall and inflow to the facility, and using the MLP method, obtained a model for which R=0.91. In contrast our results of MLSS calculation are better than those of Guclu and Dursun (2010), who using the MLP method and on the basis of Q, CODin, TSSin, L_{COD}, TKNin and DO measurements determined a model, for which R=0.88. On the other hand

better results than those obtained by the authors mentioned were obtained by us using SVM and MARS methods, for which the R-values are equal to 0.924 and 0.938, respectively.

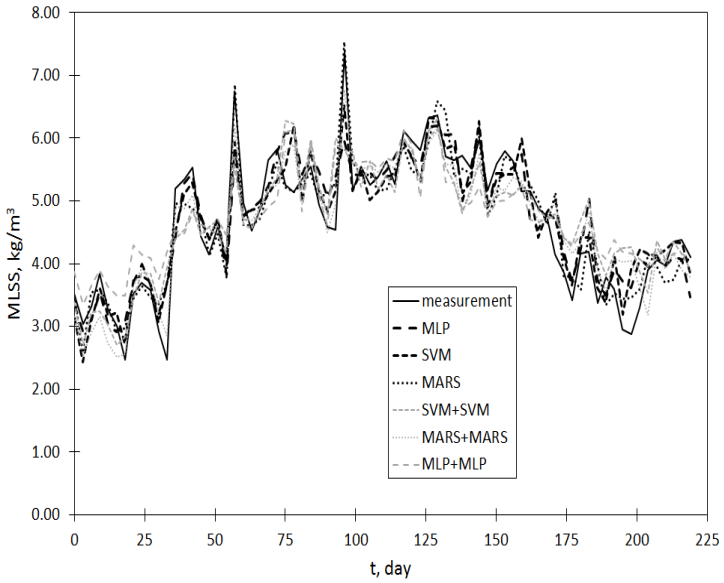


Fig. 1. Comparison of the measurements and of results of MLSS calculations done by MLP, SVM and MARS methods.

Table 3. Summary of measures to match BOD_5 calculation results to measurement data.

	Teaching				Testing			
	R	MAPE	MAE	RMSE	R	MAPE	MAE	RMSE
Method	-	%	mg/dm^3	mg/dm^3	-	%	mg/dm^3	mg/dm^3
MARS	0,91	10,99	28,53	36,72	0,92	9,68	30,71	38,76
MLP	0,72	16,57	47,34	59,67	0,72	16,20	49,27	65,34
SVM	0,79	14,28	40,64	52,45	0,81	12,57	39,61	47,52

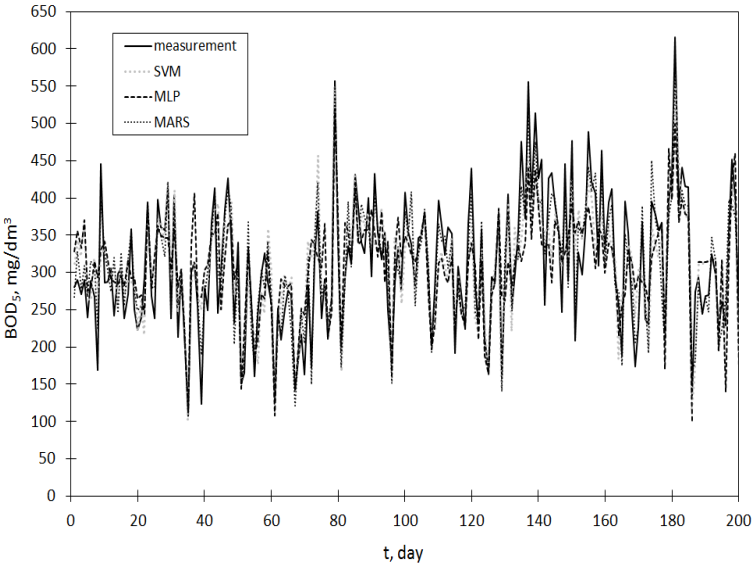


Fig. 2. Comparison of the measurements and of results of $BOD_{5,in}$ calculations done by MLP, SVM and MARS methods.

Although the MLSS modelling results obtained are correct, all the models have one basic disadvantage. Because the models include $BOD_{5,in}$ measurements, it is not possible to applicate them for the object operation in on-line mode due to the long time of performing the marking of a single value of this parameter. Therefore, based on equation (2), subsequent studies were based on determination of models for $BOD_{5.in}$ forecasting, again using the MLP, SVM and MARS methods; the values of the calculated parameters of matching the models to measurements are presented in Table 3, and Fig. 2 shows a comparison of the results of BOD_5 calculation with the measurement data.

In case of the MARS method, the lowest error values of the BOD_5 prognosis were obtained for 7 nodes, whereas in case of the MLP, the lowest error values were obtained for 8 neurons in the hidden layer and for the exponential and linear activation functions in the hidden and the output layers, respectively. The lowest error values were obtained for C=200, ε=0.01 and γ=0.20 in the case of the SVM method.
On the basis of Table 3 it can be concluded that the lowest values of errors in the $BOD_{5,in}$ forecast were obtained using the MARS method, while the highest values of errors in the forecast were obtained using the MLP method.
The R-values obtained in the study are slightly lower than those obtained by Dogan et al. (2008), who on the basis of CODin, TNin, TPin, SSin and Q measurements developed a model for the BOD_5 forecast using the MLP method, obtaining for it R=0.95. This is due to the fact that these authors used in their calculations much higher number of independent variables describing the modeled parameter BOD_5.
Since the determined models of MLSS and BOD5 parameters were characterized by satisfactory predictive capabilities, in subsequent studies the MLSS variable has been modelled using BOD_5 models, i.e. in relation (1) describing the MLSS model instead of BOD_5 measurements, the results of modelling this parameter on the basis of equation (2) were substituted; in the calculations the following combinations of methods used for the determination of the relevant models were examined: MARS+MARS, SVM+SVM and MLP+MLP. The results of the calculations are presented in Table 4 and in Fig. 1 the comparison of the calculation results with the measurements data is shown.

Table 4: Summary of measures to match MLSS calculation results to measurement data for the various combinations of the modelling methods used.

Method	R	MAPE	MAE	RMSE
	-	%	kg/m^3	kg/m^3
MLP+MLP	0,832	13,32	0,51	0,62
SVM+SVM	0,865	10,16	0,45	0,53
MARS+MARS	0,902	8,56	0,39	0,47

Analyzing the matching measure values received (Table 4), it can be concluded that the models obtained based on the combination of equations (1) and (2) predict the MLSS active sediment concentration with satisfactory accuracy,

especially in the case of using the MARS modelling method in both calculations, although the results obtained are worse than when the BOD_5 measurements were used directly in the MLSS modelling. Among the studied combinations of methods, the lowest MLSS modelling errors were obtained for the MARS+MARS combination, and the highest for the MLP+MLP combination. In addition, it can be noted that replacing the measured BOD_5 values with modelling values leads in many cases to an underestimation of the MLSS maximum values, which could potentially lead to exceeding the allowed limit values of the treated wastewater quality indicators ($BOD_{5,eff}$, COD_{eff}, TSS_{eff}, TN_{eff}, $N\text{-}NH_{4,eff}$, TP_{eff}) during the operation of the plant. Therefore, further testing is needed to improve the accuracy of the calculations of the models proposed in the study.

REFERENCES

Burges C. (1998) A Tutorial on Support Vector Machines for Pattern Recognition In: Fayyad U., Knowledge Discovery and Data Mining, Kluwer, 1998, 1–43.

Dogan E., Ates A., Yilmaz E.C., Eren B. (2008) Application of Artificial Neural Networks to Estimate Wastewater Treatment Plant Inlet Biochemical Oxygen Demand. Environmental Progress, 27(4), 439–446.

Friedman J. (1991) Multivariate Adaptive Regression Splines. Annals of Statistics 19, 1.

Güçlü D., Dursun Ş. (2010) Artificial neural network modelling of a large-scale wastewater treatment plant operation. Bioprocess and Biosystems Engineering, 33 (9), 1051–1058.

Henze M., Gujer W., Mino T., Matsuo T., Ventzel M.C., Marais G.V.R., Van Loosdrecht M. (1999) Activated Sludge Model No 2d. Water Science and Technology, 39(1), 165–182.

Hong Y.S., Bhamidimarri R. (2003) Evolutionary self-organising modelling of a municipal wastewater treatment plant. Water Research 37, 1199–1212.

Ossowski S. (2013) Neural Networks for Information Processing. Publishing House of the Warsaw University of Technology, Warszawa.

Szeląg B., Gawdzik J. (2017) Assessment of the effect of wastewater quantity and quality, and sludge parameters on predictive abilities of non-linear models for activated sludge settleability predictions. Polish Journal of Environmental Studies, 26(1), 315–322.

Szeląg B., Gawdzik J. (2017) Application of Selected Methods of Artificial Intelligence to Activated Sludge Settle-ability Predictions. Polish Journal of Environmental Studies, 25(4) 1709–1714.

Szeląg B., Barbusiński K., Studziński J. (2018) Activated sludge process modelling using selected machine learning techniques. Desalination and Water Treatment, 1–10.

Szeląg B., Studziński J. (2017) A data mining approach to the prediction of food-to-mass ratio and mixed liquor suspended solids. Polish Journal of Environmental Studies, 26(5), 2231–2238.

Szetela R., Dymaczewski Z. (2002) Modyfikacja obecnej postaci modelu osadu czynnego ASM 2d. Ochrona Środowiska, 1, 3–8.

MODELING EMERGENCE BY INTEGRATING DEVS AND MACHINE LEARNING

Daniel Foguelman, and Rodrigo Castro
Departamento de Computación
FCEyN, UBA and ICC, CONICET
Ciudad Universitaria, Pabellón 1
C1428EGA, Buenos Aires, ARGENTINA
{dfoguelman,rcastro}@dc.uba.ar

KEYWORDS
Simulation, Modeling, DEVS, Emergence, Complex Adaptive Systems, Boids, Machine Learning.

ABSTRACT

Analyzing complex adaptive systems is always a challenging task. Nature and its underlying governing rules do not always show clear patterns. The hypothesis of emergent properties in such systems is hard to formulate and difficult to infer. In this context, a great effort is being done by the Modeling and Simulation (M&S) community towards modeling and handling emergent behavior. Our research proposes minimal modifications to the Discrete Event System Specification (DEVS) M&S framework that brings the detection of emergent behavior into the loop of a DEVS simulation. New knowledge about behavior at micro levels is learned dynamically and encoded into the DEVS layered structure at macro levels. The approach bridges the gap between micro and macro representations of a given system. A proof of concept was implemented for the canonical Boids model showing promising results.

INTRODUCTION

The idea of *emergence* has been exhaustively studied by philosophy for centuries (O'Connor and Wong, 2015). Its importance on the analysis of diverse systems in natural and social sciences is undebatable. In complex adaptive systems the concept of *emergence* can be defined as the formation of order from disorder based on self-organization, or similarly, as the materialization of higher-scale properties previously unexpected while observing lower-scale behaviors. To reason about these properties it is essential to capture the underlying patterns and first principles that drive a systems' behavior. Quoting John Holland (Holland, 1998) (p.188) while reasoning about levels of description and reductionism in science, instead of holding to the notion that "all phenomena in the universe are reducible to the laws of physics" one should approach the study of systems more carefully by recognizing that "all phenomena are *constrained* by the laws of physics" and then focus into the identification and modeling of the laws that create emergence.
Several authors elaborated on the importance of computational Modeling and Simulation (M&S) of emergence (Wildman and Shults, 2018; Mittal et al., 2018) to reach a greater understanding of these attributes without assuming top-down or whole-part causes.
According to (DeLanda, 2011) simulation contributions to *emergence* helps to restore its legitimacy because they can expose interactions between virtual entities from which characteristics, tendencies, and capacities actually emerge. However, methods to explain or infer *emergent* properties that rise from interactions in a system are still missing (Szabo and Birdsey, 2017).

PREVIOUS WORK

Modeling *emergence* for simulation purposes is not new. In the last few years several authors developed new modeling formalisms that tackle this phenomenon from different angles.
M&S developments in *emergence* towards new formalisms have been developed based on Agent-based Modeling (Bouarfa et al., 2013), Grammar based models (Szabo and Teo, 2015) and DEVS (Zeigler and Muzy, 2016) just to name a few.
Other types of simulation models exist that, despite of not being designed specifically for handling *emergence*, are good candidates to model such phenomena thanks to their inherent multi-level capabilities (Steiniger and Uhrmacher, 2016) and (Uhrmacher et al., 2007).
Several examples of *emergence* simulation can also be found implemented in the Netlogo simulation environment (Tisue and Wilensky, 2004).
New challenges on where and how the community should go towards modeling *emergence* have been analyzed before. It has been put forward that one of the grand challenges on emergence is the development of "Robust Simulation Environments" that support the "simulation of [a] complex adaptive system model that manifests

accurate emergent behavior" as stated in (Diallo et al., 2018).

CURRENT LIMITATIONS AND OPPORTUNITIES

From the reviewed existing works we identified key aspects that led us to develop a novel approach to model *emergence*. These aspects are:

Maturity: recently developed modeling techniques can arguably deal with emergent behavior, their implementations are still at proof-of-concept level, and their inherent limitations are still under debate.

Usability: the design of models that can express *emergent* properties should be simple enough to concentrate the efforts more on the analysis of individuals' behavior rather than on how to translate the problem into the given formalism.

Scalability: it should be mandatory that the formalism allows for scaling the system's size so that large complex problems can be studied within reasonable simulation times.

Our assessment of the current approaches is that they present either high implementation complexities, non-formal ad-hoc designs, or too abstract formulations still far from practical applications. In this context, our driving question is: to what extent the needs raised by the M&S of emergent behavior can be met by minimal adaptations of the well known DEVS framework?
By taking this road we aim to inherit the good performance of DEVS in terms of the three aspects put forward above.

THE DEVS MODELING AND SIMULATION FRAMEWORK

The Discrete Event System Specification (DEVS, (Zeigler et al., 2000)) is a mathematical formal framework based on general systems theory for modeling and simulation of discrete, continuous and hybrid systems (Wainer and Mosterman, 2010; Cellier and Kofman, 2006). Since its first specification in 1976 (Zeigler, 1976) DEVS-based tools have been implemented in several programming languages and applied to a wide range of areas in nature, physics, engineering, computing, etc. The formal specification allows for analytic manipulation, offering hierarchical composition of structural (coupled) and behavioral (atomic) models defined by compact tuples of sets and functions.

OUR CONTRIBUTION

We propose an approach based on simple modifications over the DEVS classic formal model: to encapsulate hierarchical emergent states within DEVS layered components.

Consider the classic DEVS *Coupled* type of model CM. It consists of a module that defines *structure* of (possibly many) individual "children" models by linking their input and output ports (a directed graph where models are the nodes and links are the edges). CM also accepts external input events, emit external output events, and defines a tie-break function *select* for cases when two or more of its children need to change state at the exact same time. CM has no state of its own, and therefore no state transition function defined. However, if we consider the set of all possible combinations of states of its children models, we could reason about properties of the aggregated state set as a whole, i.e. snapshots describing some "aggregated state" of CM based certain observations made on its children states.

Following this line of reasoning we can equip the coupled model definition with two new elements: a *function* F_{CM} that computes aggregated states the constituent models, and a *coupled state* S_{CM}. The later is the result of applying F_{CM} to the state-composition of the children's states.

Children models can, in turn, be influenced by a kind of *downward causation* (actually a downward *influence*) as they are allowed (not obliged though) to access S_{CM} while computing their local state transitions.

The exchange of information between parent and children is done via the DEVS abstract simulator's standard data-transport layer.

With this approach, when aggregating the children states we have the chance to detect or infer *emergent* properties applying machine learning techniques during runtime. Children models can then eventually communicate with its parent to ask questions regarding properties of a global state which can be shaped by *emergent* behaviors.

Due to its simplicity, our solution integrates cleanly with already existent DEVS simulation engines, providing a robust platform to build upon.

With a more expressive couple model structure we provide an elegant entry point to incorporate data analysis-in-the-loop capabilities (machine learning being one specific, though powerful case).

The ease of implementation in existing DEVS-based toolkits like PowerDEVS (Bergero and Kofman, 2011) or PythonPDEVS (Van Tendeloo and Vangheluwe, 2015), integrates the idea smoothly with proven simulation solutions. This is achieved without disrupting the classic DEVS modeler's design work flow, as F_{CM} and S_{CM} can be just ignored (left undefined) without producing any impact on a classic DEVS model. Moreover, if a children model attempted to retrieve information form an undefined S_{CM} aggregated set, it should simply know how to deal with an empty response.

A PROOF OF CONCEPT CASE STUDY

A proof of concept case study is provided to demonstrate the suitability of our proposed DEVS modification. We adopt a canonical distributed behavioral model for Flocks proposed by (Reynolds, 1987). The relevance of this model lies in its capabilities to benchmark formal models targeting *emergence*. Our Flocks model is specified as follows.

We define a *Flock* DEVS coupled model that contains several birds modeled as *Bird* DEVS atomic models.

A Bird state is defined by its *direction* and a pair of coordinates x,y in a continuous 2D space.

We determine the *flockmates* of each bird as the set of birds within a given radius. Flockmates are computed accessing the states of all Bird models and applying the machine learning algorithm *Radius Neighbors Regressor*, a regression model based on neighbors within a fixed radius. The aggregated state S_{CM} is generated in simulation time by the coupled model.

Several similar functions could be encoded into the Flock coupled model to aggregate different information from its children. In this case we designed methods to give information about the closest bird, the flockmates and the average alignment and cohesion directions that a bird should follow.

The internal state transition function that define the behavior of each Bird enforces the Boids rules (Reynolds, 1987):

Separation: if *nearest neighbor* is closer than a given maximum distance.

Alignment: otherwise, steer the direction towards the *average heading* of neighboring birds.

Cohesion: finally, turn the direction towards the average position of local flockmates.

The experiment is configured with the following parameters: the size of the grid is 300 x 300 distance units, the flock size is 500 birds, the radius of visibility for each bird is 15 units and the maximum allowed distance between birds is 2 units. The position evolves over time following the updated direction with a constant module of velocity equal to 1 distance unit per time unit. The total simulation time was set to 60 time units.

Figure 1: Evolving emergent properties. Birds' orientations, positions and clusters.
Top panel: Initial state. Bottom panel: Final state, including a full motion path for bird ID 123 (in black).

RESULTS

Figure 1 shows the spatial evolution of the model in time. Ellipses enclose flockmates when three or more birds belong to the same flock. Each bird points at its own direction. The cluster analysis that results in different assignments of birds to clustered flocks is generated in simulation time.

In Figure 2 we observe the dynamics of flocking throughout the simulation. The membership to clusters fluctuates as birds cross paths with other birds. We can notice the variance of cluster sizes, going from small sets to mega-clusters of birds that can reach up to 75% of the population. More interestingly, the number of formed clusters tend to decrease in time, showing that cohesion is an *emergent* property.

By observing Figure 3 we see that the pattern of directions stabilizes. The box plot depicts how directions are initially widely spread between 0 and 365 degrees, converging afterwards to values close to 50 degrees. This is also an evidence of emerging behavior.

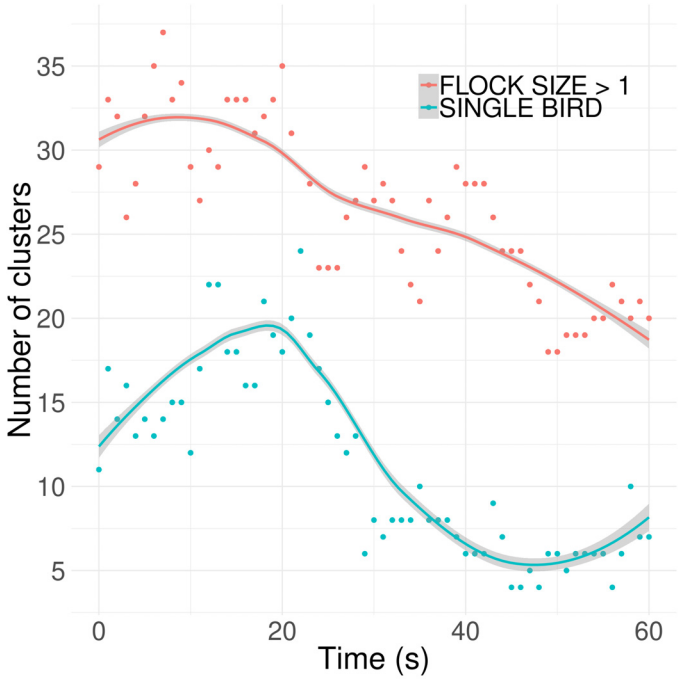


Figure 2: Emergence of clusters. Number of clusters (flock size>1) and number of single (non-clustered) birds.

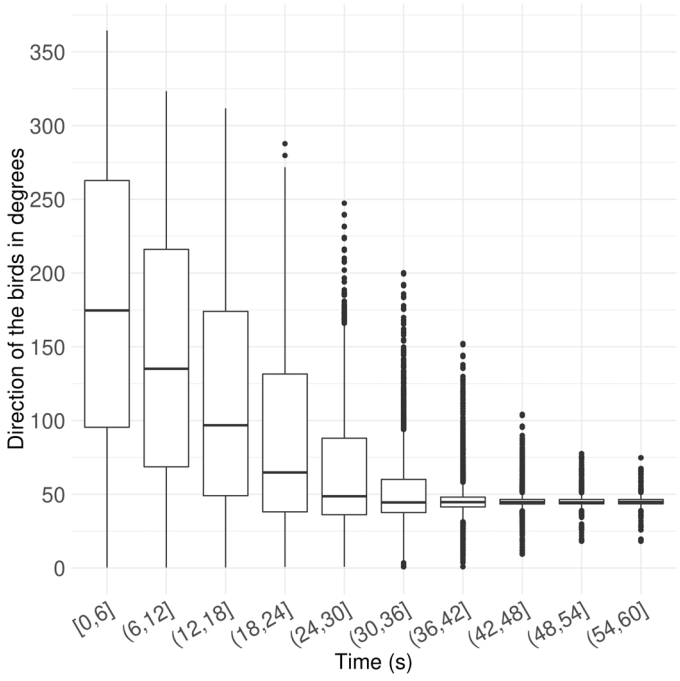


Figure 3: Emergence of coherence on the directions of birds.

CONCLUSIONS AND FUTURE WORK

Modeling and Simulating of *emergence* is a promising field. Several new approaches are rising and there are opportunities to contribute with better methods that can handle emergence with correctness, accuracy and formal soundness.

With such goals in mind we presented a modification to the DEVS formal modeling framework that inter operates transparently with already existing DEVS toolkits. The integration of machine learning algorithms that are able to predict, classify and infer *emergent* properties based on the individuals behavior is auspicious.

A case study using the well established Boids model showed that the strategy of equipping a DEVS coupled model with an aggregate state plus an analysis function is powerful enough so as to generate and handle emergent behavior.

We were able to reproduce flocking behavior that is qualitatively comparable for instance with results obtained with a similar model (Wilensky, 1998) implemented with the NetLogo tool.

As promissory as the results may appear, there is still the need to analyze the limitations of the approach. What kind of *emergent* properties can be possibly detected, what are the type of problems that this framework can model, or what is the trade-off between modeling capabilities and simulation performance? These are questions to be addressed in a future work. Moreover, to assert that our modification is still is a DEVS model requires to provide a formal proof for the very important closure under coupling property, which is part of our work in progress.

References

F. Bergero and E. Kofman. Powerdevs: a tool for hybrid system modeling and real-time simulation. *Simulation*, 87(1-2):113–132, 2011.

S. Bouarfa, H. A. Blom, R. Curran, and M. H. Everdij. Agent-based modeling and simulation of emergent behavior in air transportation. *Complex Adaptive Systems Modeling*, 1(1):15, 2013.

F. E. Cellier and E. Kofman. *Continuous system simulation*. Springer Science & Business Media, 2006.

M. DeLanda. *Philosophy and simulation: the emergence of synthetic reason*. Bloomsbury Publishing, 2011.

S. Diallo, S. Mittal, and A. Tolk. Research agenda for next-generation complex systems engineering. *Emergent Behavior in Complex Systems Engineering: A Modeling and Simulation Approach*, pages 370–389, 2018.

J. H. Holland. *Emergence: From chaos to order*. Redwood City, CA, Addison-Wesley, 1998.

S. Mittal, S. Diallo, and A. Tolk. *Emergent behavior in complex systems engineering: a modeling and simulation approach*. John Wiley & Sons, 2018.

T. O'Connor and H. Y. Wong. Emergent properties. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2015 edition, 2015.

C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 25–34, New York, NY, USA, 1987. ACM.

A. Steiniger and A. M. Uhrmacher. Intensional couplings in variable-structure models: An exploration based on multilevel-devs. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 26(2):9, 2016.

C. Szabo and L. Birdsey. Validating emergent behavior in complex systems. In *Advances in Modeling and Simulation*, pages 47–62. Springer, 2017.

C. Szabo and Y. M. Teo. Formalization of weak emergence in multiagent systems. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 26 (1):6, 2015.

S. Tisue and U. Wilensky. Netlogo: A simple environment for modeling complexity. In *International conference on complex systems*, volume 21, pages 16–21. Boston, MA, 2004.

A. M. Uhrmacher, R. Ewald, M. John, C. Maus, M. Jeschke, and S. Biermann. Combining micro and macro-modeling in devs for computational biology. In *Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come*, pages 871–880. IEEE Press, 2007.

Y. Van Tendeloo and H. Vangheluwe. Pythonpdevs: a distributed parallel devs simulator. In *Proceedings of the Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*, pages 91–98. Society for Computer Simulation International, 2015.

G. A. Wainer and P. J. Mosterman. *Discrete-event modeling and simulation: theory and applications*. CRC Press, 2010.

W. J. Wildman and F. L. Shults. Emergence: What does it mean and how is it relevant to computer engineering? *Emergent Behavior in Complex Systems Engineering: A Modeling and Simulation Approach*, pages 21–34, 2018.

U. Wilensky. Netlogo flocking model. *Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL*, 1998.

B. Zeigler. Theory of modeling and simulation. John Wiley & Sons. *Inc., New York, NY*, 1976.

B. P. Zeigler and A. Muzy. Some modeling & simulation perspectives on emergence in system-of-systems. In *Proceedings of the Modeling and Simulation of Complexity in Intelligent, Adaptive and Autonomous Systems 2016 (MSCIAAS 2016) and Space Simulation for Planetary Space Exploration (SPACE 2016)*, 2016.

B. P. Zeigler, H. Praehofer, and T. G. Kim. *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. Academic press, 2000.

HAND GESTURE CLASSIFICATION WITH USE OF CONVOLUTIONAL NEURAL NETWORKS

Tomasz Grzejszczak
Silesian University of Technology,
Faculty of Automatic Control,
Electronics and Computer Science,
16 Akademicka Street, 44-100
Gliwice, Poland
Tomasz.grzejszczak@polsl.pl

Robert Roth and Reinhard Möller
Chair of Automation / Computer Science
Univeristy of Wuppertal
Rainer-Guenter-Str. 21, 42119
Wuppertal, Germany
rroth@uni-wuppertal.de
r.moeller@uni-wuppertal.de

KEYWORDS
hand gestures, sign language, convolutional neural networks, deep learning, CNN, HCI, HMI

ABSTRACT

The preliminary research on stationary sign language hand gesture recognition with use of convolutional neural networks is presented. Three tests are presented that differ in number of gesture types, that is, by classes amount. The main question is, whether it is possible to use convolutional neural networks and deep learning approach in stationary gesture recognition, when there is small number of training data, what is the influence of similar gestures on classification process and is it necessary to use controlled background. Test results are provided in form of confusion matrices. Finally, the direction of future work and further research is discussed.

INTRODUCTION

Gesture recognition is widely researched. There is a constant need of improvement in the way of communication between human and machine. There is a research field called Human Computer (or Machine) Interface (HCI or HMI), where the possible interactions are examined. People tend to communicate with computer with most intuitive way.

People communicate with other people verbally and non-verbally. Among the nonverbal communication channels, one of them are gestures and in particular, hand gestures. Moreover, hand gestures and sign language is the main channel of communication for deaf people.

This paper presents the preliminary studies that should prove the concept of stationary hand gesture recognition by gesture type classification with use of Convolutional Neural Networks and Deep Learning.

GESTURE RECOGNITION

In the ongoing research on hand detection and hand landmarks localization the traditional, 3 stage image processing approach was used. The developed algorithms were divided into stages [3]:

1. Segmentation / preprocessing
2. Feature extraction
3. Gesture recognition

The first step was to determine region of the hand. Usually skin segmentation algorithms are used, however a common practice is to omit this step and assume constant background that can be easily subtracted. After preprocessing, the output is a hand mask, that is a binary image with information where is the hand located on the input image or video frame. Next, in feature extraction different algorithm can be used to find a set of most valuable features and their values. This information is used to define and construct classification method in order to output the final description of hand gesture in analyzed image.

Usually, there are two types of appliances. First, the classification of gesture that can be used for example to create a sign language translator [1]. Second, to track hand features for intuitive control [12, 2, 4]. Convolutional Neural Networks can be applied also for skeleton fitting and feature tracking [5], however the most common appliance is to classify images, thus this approach is described in this paper.

The aim of the tests is to recognize the presented gesture. User shows a hand gesture from a sign language and the hand gesture is captured with use of a digital camera. Thus, the input of the system is a hand image and the expected output is an information what gesture had been presented. This is a classification problem.

CONVOLUTIONAL NEURAL NETWORK

The idea behind use of convolutional neural network for image classification had its origin in 2012 in ImageNet competition [8]. It was the first approach that had broken the standard image processing approach. The deep neural network was created and it was trained with thousands of raw, not processed color images. No additional rules were introduced by the developer. During the training of this neural network, the neuron weights were set to automatically determine the best features that classifies the input images. Usually, the convolutional neural networks are built in form of layers. There is a mix of convolutional, sampling and classification layers, each differ in input and output size.

From this date, Convolutional Neural Networks are much more often applied to different research fields and they differ mostly in model structure. Despite the traditional image classification, convolutional neural networks are used for stereo matching [16], where the patches from two side by side cameras are located to find a depth image. When the backward learning is applied to a convolutional neural network, one can introduce a semantic matching for creating

the exact object mask [10]. Despite the image processing, deep learning can also be find in application to biomedicine [14] or in data processing for activity recognition [7, 15]. In term of human and gesture recognition, the main field of application is for face recognition [9]. There are also many publications in field of hand gesture recognition. Some of them uses depth sensor data [11], some of them are focused on dynamic gesture recognition [12]. There are also similar research in field of stationary hand gestures on color images [6, 13] that would give a great opportunity to inspire the future work and compare the results as soon as the research will finish the preliminary stage.

HAND GESTURE RECOGNITION DATA BASE

Hand Gesture Recognition database is available online (link omitted for double blind review). It has been created for research on algorithms for hand landmarks detection and localization with use of image processing. Most important research concerned skin region, wrist point and finger points localization and detection.

Each image in database is equipped with curtail landmarks description and localization and skin region mask. However, this information is not useful in deep learning approach.

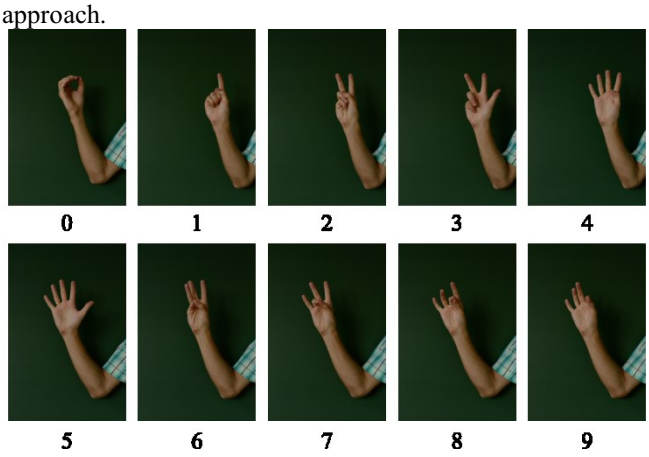


Fig. 1 Exemplary images from HGR dataset from one person. The presented gestures correspond to numbers from American Sign Language

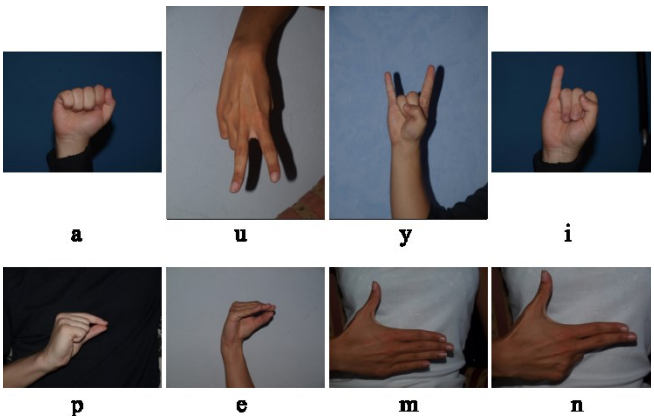


Fig. 2 Exemplary images from HGR test dataset. The presented gestures corresponds to letters from Polish sign language

Most important thing is that images were taken in different conditions, that is with and without controlled light and background. There are many individuals performing the same gesture, that ensures the variety of gestures presenting understanding and lowers the biometry factor impact. Each image is described according to person ID, presented gesture meaning and gesture association (Polish or American Sign Language). In this paper the parts of this dataset are used. Each test describes the size and type of used dataset.

The main drawback of this set is that it is relatively small to use it in deep learning approach. It is enough to use this set for traditional image processing approach. This article presents the prove of concept, whether it is necessary to extend the database.

TESTS

The size of input images are down sampled to maximum of 256 pixels height or width.

In order to provide more samples to train the neural network, each image is ten times randomly reshaped, zoomed, rotated by a random factor and/or horizontally flipped to simulate a left hand gesture.

The convolutional neural network uses a sequential model. There are 3 layers: convolutional, rectified linear unit (ReLU) type activation and maximum puling that are repeated 3 times with different parameters. Then, the last layers are a combination of flattering, dense and dropout. The final classification is performed with use of softmax classifier. The neural network is trained with use of binary crossentropy loss function. The neural network structure is presented in figure 3. The convolutional neural network model structure is chosen intuitively. A part of future work focusses on determining a proper and optimized model structure.

In each test, the available images were divided into 3 sets: training, validation and test. For each gesture, the presenting individual is randomly assigned into one of 3 sets, however the set size is fixed. For example, in first test, there are 18 individuals presenting 6 gestures, that gives different 108 images. There are 18 images of 18 individuals presenting gesture “1”. From this set 13 images are randomly assigned to training set, 4 images are used for validation and 1 for testing.

6 gestures

The first test was the easiest one, containing 6 gesture types. Depending on the count of risen fingers, the presented gestures corresponded to gestures meaning 0-6 from American Sign Language. Figure **1** presents the representation of digits in American Sign Language presented by one person. In this test only images 0-6 presented by 18 individuals. The number of risen fingers directly indicates the gesture type, thus this is why this test is called the easiest. The task of counting risen fingers has been examined many times with use of multiple approaches. [ref]
Moreover, to make it even easier, the constant background is assumed.

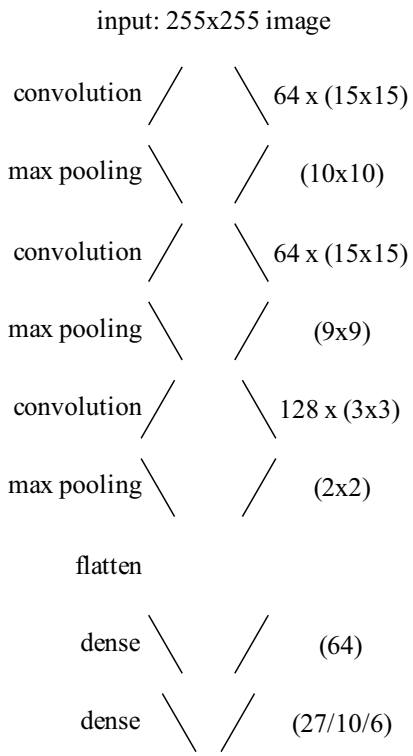


Fig. 3 Model structure

In this test there are 18 individuals presenting 6 gestures. Each image is repeated 10 times with small transformation that gives total 1080 samples, 180 for each class. The images are divided into training, validation and test sets in proportion 13:4:1.

Table 1 Confusion matrix of 6 gestures test

	0	1	2	3	4	5
0	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1	0.0018	0.9891	0.0058	0.0018	0.0014	0.0000
2	0.0020	0.0632	0.9170	0.0072	0.0106	0.0000
3	0.0000	0.0000	0.0001	0.9575	0.0000	0.0423
4	0.0039	0.0178	0.1123	0.0115	0.8393	0.0152
5	0.0000	0.0000	0.0000	0.0001	0.0004	0.9995

The test outcome is presented in Table 1. Each row presents the probability of assigning the image into one of 6 classes. Class names are given in first row and first column. The intensity of background color indicates the certainty value. From Table 1 one can read that each gesture is properly classified. The best classification, with 100% accuracy was for the gesture "0" with no shown fingers. The worst classification is for gesture "4" that can be mistaken for gesture "2". The reason for this score is that the individual presented this gesture not precisely and the fingers were touching. The overall certainty of this test is about 95% that is very high. This proves that this method have a potential and can used among other methods based on risen finger counting.

10 gestures

The next test is more complex. This time, the whole gesture set presented in figure **1** is used. Again, there are 18 individuals and the images set is divided into training, validation and test sets in proportion 13:4:1. The total number of images is 180. The problem is that gestures 6-9 are similar. The task of classification cannot be done by risen finger counting, because each tine 3 fingers are risen and the difference is whether the thumb is touching index, middle ring or pinkie finger.

Table 2 Confusion matrix of 10 gestures test

	0	1	2	3	4	5	6	7	8	9
0	1.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
1	.00	.23	.10	.02	.02	.00	.07	.01	.47	.06
2	.01	.13	.56	.00	.01	.00	.10	.02	.14	.03
3	.00	.00	.00	.57	.00	.41	.00	.00	.01	.01
4	.02	.04	.28	.02	.09	.01	.33	.01	.14	.06
5	.00	.00	.01	.07	.02	.26	.14	.13	.04	.33
6	.02	.02	.05	.01	.24	.03	.01	.23	.33	.07
7	.01	.03	.24	.01	.17	.03	.10	.03	.34	.04
8	.01	.01	.02	.00	.02	.01	.02	.57	.23	.11
9	.00	.04	.11	.03	.21	.10	.24	.02	.05	.18

The test outcome is presented in Table 2. Each row presents the probability of assigning the image into one of 10 classes. Class names are given in first row and first column. The intensity of background color indicates the certainty value. From Table 1 one can observe that the most certain gesture is the one without risen fingers, that is "0". The classification of this gesture is with 100% certainty and other gestures is believed to be "0" with not more than 2%. The problem is with other gestures. Only "2" and "3" are marginally classified correctly. Gestures 6-9 are not only similar to each other, but also to gestures 1-5 making the whole classification wrong. It is most likely possible that the initial image size after down sampling is too small. It is very hard to determine the position of fingers. This test is not a success, however is shows the most important problems of similar gesture classification.

27 gestures

Finally, the full test is performed on full HGR dataset that is a set of all stationary gestures from Polish Sign Language. 12 individuals are presenting 27 different gestures. Sometimes one individual is presenting the gesture several times, making the total of 33 images per one gesture. The images are divided into training, validation and test classes in proportion 27:5:1.

Fig. 2 presents the part of the test set. It can be observed that there are random individuals in each class. The background is rather uniform, but not strictly controlled. Looking at Fig. 2 one can notice some similarities of some gestures and uniqueness of others. This observation can be confirmed in test results presented in Table 3. Gestures "a", "u" and "y" are unique. Previous tests showed that fist gesture ("0" or "a") is the most unique. Here, also "u" and "y" are unique, because one is the only gesture upside down and the second require wide spacing between index and

Table 3 Confusion matrix of 27 gestures test. Numbers in table are in percent of classification certainty.

	1	2	3	4	5	a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	r	s	t	u	w	y	z
1	28	09	04	00	00	00	00	01	01	00	00	00	00	00	03	32	04	02	01	00	10	01	01	00	00	00	03
2	01	01	01	04	01	02	00	03	03	07	00	00	00	01	00	03	34	27	00	08	01	00	00	00	00	01	00
3	28	30	30	01	04	00	00	00	00	00	00	00	00	00	00	04	02	01	00	00	00	00	00	00	00	00	00
4	01	03	03	48	05	00	00	02	01	03	00	00	00	00	00	01	18	09	01	03	00	00	00	00	00	00	00
5	02	01	01	08	03	04	03	08	07	05	06	08	03	03	01	02	02	02	08	05	01	06	05	00	01	02	02
a	00	00	00	00	00	98	00	00	00	01	00	00	00	01	00	00	00	00	00	00	00	00	00	00	00	00	00
b	00	00	00	00	00	00	24	00	00	00	00	00	01	05	37	03	00	00	00	00	24	00	00	00	01	00	04
c	00	00	00	01	00	15	00	09	04	29	03	02	01	03	00	00	02	04	01	20	01	00	01	01	00	02	00
d	00	00	00	00	00	00	00	45	07	13	13	06	00	00	00	00	00	00	04	08	00	00	03	00	00	00	00
e	00	00	00	00	00	02	00	12	08	36	01	07	00	00	00	00	00	00	00	33	00	00	00	00	00	00	00
f	00	00	00	00	00	00	00	02	01	00	18	07	00	00	00	00	00	00	27	00	00	20	25	00	00	00	00
g	00	00	00	00	00	00	00	07	02	00	10	44	01	01	00	00	00	00	05	00	00	06	22	00	00	00	01
h	00	00	00	00	00	00	01	11	03	01	04	06	16	07	01	01	00	00	01	00	01	02	21	00	03	17	04
i	00	00	00	00	00	27	03	02	02	04	02	05	04	28	01	00	00	00	00	01	01	00	01	00	01	17	01
k	00	00	00	00	00	02	02	18	09	04	17	17	05	04	01	00	00	00	01	04	01	01	09	00	00	02	02
l	00	00	00	00	00	01	04	01	01	00	01	00	01	06	15	21	00	00	05	00	10	07	05	02	02	00	17
m	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	84	16	00	00	00	00	00	00	00	00	00
n	00	00	00	01	00	00	00	00	00	00	00	00	00	00	00	00	84	15	00	00	00	00	00	00	00	00	00
o	00	00	00	00	00	00	03	05	03	01	15	06	04	02	03	01	00	00	12	01	02	13	22	00	02	00	05
p	00	00	00	00	00	06	00	04	02	74	00	00	00	01	00	00	00	00	00	12	00	00	00	00	00	01	00
r	00	00	00	00	00	00	01	00	00	00	00	00	06	01	24	15	00	00	00	00	13	03	07	00	03	00	26
s	00	00	00	00	00	00	01	01	01	00	04	04	10	03	03	00	00	00	01	00	01	04	41	00	13	02	11
t	00	00	00	00	00	00	02	01	01	00	07	03	11	06	03	00	00	00	00	00	03	01	30	00	07	02	22
u	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	99	00	00	00
w	00	00	00	00	00	00	10	00	00	00	00	00	12	05	28	01	00	00	00	00	08	00	02	00	20	04	08
y	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	99	00
z	00	00	00	00	00	04	03	19	11	05	09	02	02	05	01	01	00	00	08	04	03	02	11	04	00	01	03

pinkie fingers. “I” was classified marginally correct and according to confusion matrix, it can be mistaken with “a” or “y”. Those gestures are similar and differ in rise of index or pikey finger. Other similar gesture pairs are “p” – “e” and “m” – “n”. Those gestures differ only in small detail and those properties can be observed in confusion matrix.

The overall number of positively classified gestures is 10 out of 27 (37%). If we allow the answer to be an alternative of two best choices (for example treat the answer “gesture ‘e’ is ‘p’ or ‘e’ “ as correct) then the overall score would be 16 out of 27 (60%). This is still a small number, however it is big enough to prove the concept of this approach.

It is also worth noticing, that gestures “p” and “e” differ in background and elbow location, and their similarities lie only in finger arrangement. The idea of convolutional neural networks guarantee that the irrelevant features are disregarded. In this case, when different individuals are randomly assigned to different classes for different gestures, the individual features, such as hand orientation, background color or shirt color are not considered.

CONCLUSIONS

The following research is the result of preliminary studies. This is the proof of concept of gesture classification with use of deep learning and convolutional neural networks. The main problem in this approach is the small data set for training of convolutional neural networks.

Three tests are presented. An easy test with 6 easy to distinguish gestures gives all correct classification with high certainty. Other tests proved that adding similar gestures are hard to distinguish but also it worsen the already formed rules for previously distinguishable gestures. The overall score proves the concept of gesture classification with use of convolutional neural networks. Moreover, the last test showed that the approach is not highly sensitive on image background. The provided scores are low, however expanding the gesture image data set and expanding the training set can help train the neural net, which will improve the classification score.

FUTURE WORK

There are two types of studies that will be held in future. First, there is a need of better understanding of neural network structure. The parameters should be understood more precisely and the neural network structure should be chosen in such way that is maximizes the classification score. Different model structures will be proposed and different parameters will be examined. Also the size of input image after downscaling is important. It was observed that test with 10 gestures could give bad results because of too huge downscaling, making gestures images hard to distinguish for human.

On the other hand, deep learning approaches require a huge training set. Usual training sets are counted in thousands. Here the training set is 13-27, which after artificial enhancement gives 130-270. Definitely, the HGR dataset should be expanded and the influence of extended training set can be examined.

ACKNOWLEDGEMENTS

The research was supported by Polish National Science Center under grant 02/010/PBU17/0090 (PBU/29/RAu1/2017/505). The research presented here was supported by Polish Ministry for Science and Higher Education for Institute of Automatic Control, Silesian University of Technology, Gliwice, Poland under internal grant 508/RAU1/2017 t.31. The calculations were performed with the use of IT infrastructure of GeCONiI Upper Silesian Centre for Computational Science and Engineering (NCBiR grant no POIG.02.03.01-24-099/13).

REFERENCES

[1] Zhiwu Huang, Chengde Wan, Thomas Probst, and Luc Van Gool. Deep learning on lie groups for skeleton-based action recognition. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6099–6108. IEEE computer Society, 2017.

[2] Vijay John, Ali Boyali, Seiichi Mita, Masayuki Imanishi, and Norio Sanma. Deep learning-based fast hand gesture recognition using representative frames. In *Digital Image Computing: Techniques and Applications (DICTA), 2016 International Conference on*, pages 1–8. IEEE, 2016.

[3] Kathrin Kalischewski, Daniel Wagner, Jorg Velten, and Anton Kummert. Activity recognition for indoor movement and estimation of travelled path. In *Multidimensional (nD) Systems (nDS), 2017 10th International Workshop on*, pages 1–5. IEEE, 2017.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[5] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, and Gang Hua. A convolutional neural network cascade for face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5325–5334, 2015.

[6] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[7] Pavlo Molchanov, Shalini Gupta, Kihwan Kim, and Jan Kautz. Hand gesture recognition with 3d convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1–7, 2015.

[8] Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4207–4215, 2016.

[9] Oyebade K Oyedotun and Adnan Khashman. Deep learning in vision-based static [1] Susan Goldin-Meadow and Diane Brentari. Gesture, sign, and language: The coming of age of sign language and gesture studies. *Behavioral and Brain Sciences*, 40, 2017.

[2] T. Grzejszczak, A. Łegowski, and M. Niezabitowski. Application of hand detection algorithm in robot control. In *2016 17th International Carpathian Control Conference (ICCC)*, pages 222–225, May 2016.

[3] Tomasz Grzejszczak, Michal Kawulok, and Adam Galuszka. Hand landmarks detection and localization in color images. *Multimedia Tools and Applications*, 75(23):16363–16387, 2016.

[4] Tomasz Grzejszczak, Micha Mikulski, Tadeusz Szkodny, and Karol Jedrasiak. Gesture based robot control. In *International Conference on Computer Vision and Graphics*, pages 407–413. Springer, 2012.

[5] Zhiwu Huang, Chengde Wan, Thomas Probst, and Luc Van Gool. Deep learning on lie groups for skeleton-based action recognition. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6099–6108. IEEE computer Society, 2017.

[6] Vijay John, Ali Boyali, Seiichi Mita, Masayuki Imanishi, and Norio Sanma. Deep learning-based fast hand gesture recognition using representative frames. In *Digital Image Computing: Techniques and Applications (DICTA), 2016 International Conference on*, pages 1–8. IEEE, 2016.

[7] Kathrin Kalischewski, Daniel Wagner, Jorg Velten, and Anton Kummert. Activity recognition for indoor movement and estimation of travelled path. In *Multidimensional (nD) Systems (nDS), 2017 10th International Workshop on*, pages 1–5. IEEE, 2017.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[9] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, and Gang Hua. A convolutional neural network cascade for face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5325–5334, 2015.

[10] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[11] Pavlo Molchanov, Shalini Gupta, Kihwan Kim, and Jan Kautz. Hand gesture recognition with 3d convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1–7, 2015.

[12] Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4207–4215, 2016.

[13] Oyebade K Oyedotun and Adnan Khashman. Deep learning in vision-based static hand gesture recognition. *Neural Computing and Applications*, 28(12):3941–3951, 2017.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[15] Daniel Wagner, Kathrin Kalischewski, Jorg Velten, and Anton Kummert. Activity recognition using inertial sensors and a 2-d convolutional neural network. In *Multidimensional (nD) Systems (nDS), 2017 10th International Workshop on*, pages 1–6. IEEE, 2017.

[16] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32):2, 2016. hand gesture recognition. *Neural Computing and Applications*, 28(12):3941–3951, 2017.

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[11] Daniel Wagner, Kathrin Kalischewski, Jorg Velten, and Anton Kummert. Activity recognition using inertial sensors and a 2-d convolutional neural network. In *Multidimensional (nD) Systems (nDS), 2017 10th International Workshop on*, pages 1–6. IEEE, 2017.

[12] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32):2, 2016.

THE PROBLEM WITH ASYNCHRONOUS UPDATING

Joseph Kehoe
Institute of Technology, Carlow
email: joseph.kehoe@ITCarlow.ie

KEYWORDS
Asynchronous Updating, Synchronous Updating, Agent Based Models

ABSTRACT

Asynchronous updating is used in Agent Based Models and Agent Based Social Systems in preference to synchronous updating. It has been long known that the same simulation can have different outcomes when different updating strategies are used. This means that results may be "artefacts" of the specific updating strategy employed. It is shown here that asynchronous updating breaks the assumption of bounded rationality and leads to inconsistent transmission of information across simulation spaces. An analysis demonstrates that simulations with high population densities (such as Cellular Automata) accentuates this effect. These results help explain previous findings comparing different updating strategies. It also calls into question the preference for using asynchronous updating instead of synchronous updating.

INTRODUCTION

A simulation is the imitation of a system over time. Development of a simulation is dependent on having a model of the system being simulated where this model represents the key characteristics or behaviours of the system. An Agent Based Model (ABM) is one where individuals within the system and their interactions are explicitly represented. Within ABM updating is either Synchronous (SU) or Asynchronous (AU). (Huberman and Glance 1993) distinguishes between these approaches.
The SU approach is defined as having a global clock that synchronises the updating of all agent states so that all updates occur in unison. AU, on the other hand, has no global clock and updates of agent state occur in some defined sequential order. It has been demonstrated that SU and AU implementations of the same ABMs can result in widely differing behaviours. It is generally assumed that real world systems are asynchronous and AU correlates more closely with the reality they are attempting to simulate[1].

[1]We define accuracy in terms of how closely a simulation replicates the real-world system being modelled. For a simulation to be accurate anything that is not allowed within the system we are modelling must also be excluded from the simulation.

Although we are concerned here with ABM the issues concerning SU and AU implementations are also applicable to Discrete Event Simulations (DES). (Brailsford 2014) states that DES is a subset of ABM while (Onggo 2010) claims that any ABM can be translated into an equivalent DES. In any case the two approaches have much in common (Siebers et al. 2010) and the issue of SU or AU implementation of events is relevant to both approaches. This paper shows that although AU and SU approaches can give different results, it has not been conclusively demonstrated that the asynchronous approaches are the more accurate. Flaws in the approach to implementing asynchrony advocated in (Huberman and Glance 1993) are outlined.
In the next section the differences between the synchronous and asynchronous approaches are outlined. Following this the problems associated with AU are examined. Finally the conclusion lists a summary of the arguments in favour of a synchronous approach.

SYNCHRONOUS AND ASYNCHRONOUS UPDATING

In the paper *Evolving games and computer simulations* (1993) a case was made for an asynchronous approach to real world simulations. Synchronous simulations are defined as those in which all the agents in that simulation are updated simultaneously and instantaneously at each time step. Each step is a discrete quantum of time and the simulation progresses as a sequence of discrete states, one per time step. The state at step n is dependent solely on the state at step $n-1$. The best known simulation of this type is Conway's Game of Life (Gardner 1970).
It is argued (Huberman and Glance 1993, Caron-Lormier et al. 2008, Cornforth et al. 2005) that natural social systems belong to the class of asynchronous systems where there is no "global clock" that causes all agents to update their state in unison. These papers cite, in support of this, the fact that agents in a social setting act at different and uncorrelated times on the basis of information that may be imperfect and delayed. Using SU, Nowak developed a simulation of Spatial Iterated Prisoners' Dilemma (Nowak and May 1992) and showed that the simulation generates chaotically chang-

ing spatial patterns, in which *cooperators* and *defectors* both persist indefinitely. (Huberman and Glance 1993) then used the AU approach on the same simulation to instead show that the simulation always evolves, within 100 generations into a steady state where all agents become *defectors*. Thus we have two clearly contradictory results deriving from the application of the same rule and differing only in updating technique.
The conclusion drawn by (Huberman and Glance 1993) is that to mimic continuous real world systems we need a procedure that ensures the updating of interacting agents is continuous and asynchronous. AU is implemented by:

> choosing an interval of time small enough so that at each step at most one individual agent is chosen at random to interact with its neighbours. During this update, the state of the rest of the system is held constant. The procedure is then repeated throughout the array for one player at a time, in contrast to a synchronous simulation in which all the agents are updated at once. (Huberman and Glance 1993)

This finding has been backed up since then by numerous studies comparing synchronous and asynchronous Cellular Automata (Caron-Lormier et al. 2008, Schönfisch and de Roos 1999, Cornforth et al. 2005, Ruxton and Saravia 1998). It has become standard practice to use AU for Agent Based Models (ABM) and in particular Agent Based Social Simulations (ABSS). AU has been adopted by the major agent toolkits such as NetLogo, Repast, Mason and Swarm(North et al. 2013, Luke et al. 2005, Berryman 2008). Standard ABSSs, such as the canonical Sugarscape (Epstein and Axtell 1996), assume an AU implementation. The ability to execute a sequence of agent actions in a random order, an essential part of asynchronous simulations, forms part of *StupidModel* (Railsback et al. 2005), a suite of models designed to test the suitability of any toolkit for ABM development.
For AU to work each action has to be instantaneous in time (Fatès 2013), that is each action has no duration, otherwise it makes no sense to say that the time interval is fixed in size yet any number of sequential actions can take place within this same time interval.
Some researchers assume that real world systems are asynchronous in nature (Caron-Lormier et al. 2008, Cornforth et al. 2005). Rather interestingly, there does appear to be consensus that the order of the updates is important yet debate remains as to which is best (Ruxton and Saravia 1998).
The preference for AU has been accentuated by a lack of synchronous algorithms that can handle the complex interactions that occur in ABM. This contrasts with Cellular Automata (CA) based simulation where, because the interactions are simpler SU algorithms exist. For example, CA-based Simulations of traffic flow, a real world system, employ SU (Burstedde et al. 2001) as standard. It is not uncommon to see CA based simulations employ both updating methods and comparisons made from the results as in (Bezbradica et al. 2014, Bach* et al. 2003). Even in CA, Grilo and Correia (2011) states that the effects of the two different strategies are not well understood.
It is our contention that the case for choosing AU over SU in ABM has not been convincingly made and given that it is accepted that the outcomes of a simulation can depend on the approach taken we need to reexamine the preference for the AU and exclusion of SU. In this paper it is demonstrated that AU violates *Bounded Rationality*, an important property of Agent Based Social Simulations.

FLAWS IN ASYNCHRONOUS UPDATING

Bounded rationality is the idea that in decision-making rationality of individuals is limited by the information they have available to them, the cognitive limitations of their minds, and the finite amount of time they have to make a decision. Many consider this an essential property within ABSS Epstein (1999). Within ABM this principle is generally taken to mean that agents do not have access to global information. This principle of bounded rationality is enforced in ABM by ensuring that *an agent can only be aware of the state of other agents within its neighbourhood (locality).* This places limits on the information agents have available at any one time. They can see their locality as it is now and gain information about the world outside their locality only through time delayed information transfer (e.g. gossip).
Locality is, of course, defined in a simulation specific way. In some cases locality resembles physical proximity in the real world but in others (for example, simulating Facebook or Twitter connections to study how memes traverse social networks) it does not. Here locality is defined information theoretically. Anyone who I receive information from in real time (instantaneously) is in my "locality" or "neighbourhood". If I am undertaking a video conference call with colleagues then the people physically present in the room with me as well as the people on screen (even though they are physically removed from me) are in my locality but everyone else outside of the room is not.
A local neighbourhood may change from step to step but within each time step it is fixed. The results of actions taken during the current step can change the neighbourhood of an agent for the following step. This principle guarantees that an agent cannot be omniscient, that is, an agent cannot be aware of the global state as it is *now* or have complete knowledge of the universe. If an agent is not in my immediate neighbourhood then I cannot know what it is doing now. I may find out at a later time what it was doing through time delayed

information diffusion (e.g. gossip) but that is time delayed information. SU guarantees bounded rationality by imposing a consistent speed for the transmission of information across a simulation space. If this property is important in a particular ABM then more consideration should be given to using SU.
Take as a *gedankenexperiment* or thought experiment the case of a simulation where agents perform some action as soon as they become aware that some particular agent A is dead. An agent B can become aware of this fact under two conditions:

1. If B is a neighbour of A and witnesses A's death;
2. If B is not a neighbour of A but one of B's neighbours is aware of A's death and informs B.

Now once A dies the amount of time steps it takes for any other agent to find out should vary in direct proportion to the distance that agent is from A. Agents in the immediate locality of A should be first to know (as they witness A's death) and this information should then percolate through the system from local neighbourhood to local neighbourhood over one or more subsequent steps. Under SU this is exactly what happens. Now consider what happens under AU if we are an immediate neighbour of A. If A dies in step i when do we find out? It depends on the order in which the actions occur *during* the step. If A's behaviour is scheduled under AU to occur before ours in this step then we will be aware of A's death within this step (instantaneously). However, if A's behaviour is scheduled after ours then we will not find out until the following step. On average half of A's neighbours will be aware of its demise immediately and half will not!
This random sequencing of actions means that agents **not** within A's neighbourhood can also be aware of A's change of state instantly (within the same time step). We can construct a specific sequence of an update ordering $[a_1, a_2, ..., a_n]$ where each agent in the sequence a_i runs before a_{i+1} within this step and every a_i is a neighbour of a_{i+1}. If a_1 dies then every agent in the chain will pass this information on to their neighbour within the same step. This gives a_n instant access to information from outside of its neighbourhood.
There are two related issues here:

1. Agents who are not within the locality of A can be immediately aware of the change of state of A thus violating bounded rationality;
2. Agents further away from A can be aware of A's change of state before agents closer to A are (inconsistent speed of information transfer through the system).

It is clear that instantaneous information transfer across local boundaries (we term such events *leakages*) can be caused by AU but to determine how much this can affect simulations we need to know how often such leakages occur.

PROBABILITY OF INFORMATION LEAKAGE ACROSS LOCAL BOUNDARIES

For a leakage to occur a chain of three (or possibly more) agents must be updated in an order that allows information to move across neighbourhood boundaries in a single timestep. The amount of leakage will be directly proportional to the number of such chains occuring during each timestep.
To help calculate the liklihood of leakages occuring we make some basic assumptions:

- The lattice in which the ABM runs is a torus structure. That is, it wraps around the edges. This is not uncommon in general. This asumption makes the following calculations simpler but the use of a non torus structure does not have a major effect on the outcome of the calculations;
- The locality of an agent is determined by a range R. That is, all locations within R steps of the current location are within its locality. We also assume that $R < N/2$ where N is the dimension of the simulation lattice. This is true in all but the most trivial ABM;
- Agents use a *von Neumann* neighbourhood when calculating locality. In a one dimensional lattice the neighbourhood extends to the left and the right. In a two dimensional lattice the neighbourhood extends in four directions (left, right, up and down);
- Agents are distributed evenly throughout the lattice;

We will return to reexamine the implications of the last two assumptions after we make our calculations.

One Dimensional Lattices
To begin with we will limit ourselves to one dimensional lattices. Each location has neighbours only to its left and right. Leakage in this case can only occur between a minimuim of three locations where these locations have the following properties:

- The first location, A, is a neighbour of the second location B;
- The second location B is a neighbour of the third location C;
- The third location C is **not** a neighbour of A.

We will label any three locations that satisfies these properties a *Leakable Triple*. Such a chain of locations will be updated under AU in one of the following orders: *A B C; A C B; B A C; B C A; C A B; C B A.*

Two of these six possibilities result in leakage (sequences 1 and 6) while the other four of the six possibilites (sequences from 2 to 5) do not. In a one dimensional lattice of N locations we can determine the number of such chains as follows:
We first need to enumerate the total number of possible *leakable triples* in the lattice. Then we need to calculate how many of these possible leakable triples have all three locations occupied by agents. Finally, we need to calculate how likely each occupied leakable triple is to be updated in an order that causes leakage. This, as we have seen above, is $\frac{2}{6}$ or 33%.
Every leakable triple has a rightmost location defined by the lattice dimensionality. Therefore if we can calculate the number of leakable triples that have a location L_i as a rightmost location then we can use this to calculate the total number of leakable triples on the lattice (by multiplying the number of leakable triples with a fixed rightmost location by the total number of locations in the lattice).
For any starting location A there will be R locations within its locality (or neighbourhood) to the left.

- The first such location (call it B) is one step away and will only have one other location (call it C) within its range but outside of A's neighbourood;
- The second location to the left of A is two steps away and this location will have two other locations within its range and outside of A's neighbourhood;
- The third location to the left of A is three steps away and this location will have three other locations within its range and outside of A's neighbourhood;
- The i^{th} location ($i < R$) is i steps away and this location will have i other locations within its range and outside of A's neighbourhood;
- The R^{th} location is R steps away and this location has R other locations within its range and outside of A's neighbourhood.

In total we can see that this gives a total of $1 + 2 + ... + i + ... + R$ (or $\Sigma_{i=1}^{i=R} i$) chains for this one rightmost location. In general terms we can see that this gives a total of $\frac{R(R+1)}{2}$ leakable triples with the same rightmost location. Since there are N locations in the lattice the total number of chains must be N times this number:

$$N \times \frac{R(R+1)}{2} \tag{1}$$

Each chain is relevent only if all three locations contain agents. If there are A agents on a lattice of size N then the probability of a location being occupied is $\frac{A}{N}$. We let P represent this probability of a location containing an agent, $P = \frac{A}{N}$. For CA, it is always the case that $P = 1$ as each location is also an agent ($A = N$) but for an ABM this figure will be lower as the number of agents will be less than the number of locations. The probability of all three locations within a chain containing agents is P^3. Therefore the number of occupied chains is this times the total number of possible leakable triples:

$$P^3 \times N \times \frac{R(R+1)}{2} \tag{2}$$

We know that the possibility of AU updating such a chain in a way that preserves locality is $\frac{2}{3}$ so the probability of locality being preserved throughout the entire lattice is:

$$(\frac{2}{3})^{P^3 \times N \times \frac{R(R+1)}{2}} \tag{3}$$

It immediately follows that the possibility of leakage is then given by equation 4:

$$1 - (\frac{2}{3})^{P^3 \times N \times \frac{R(R+1)}{2}} \tag{4}$$

As R and P increase in value to the probability of leakage reaches 1 *during each step.*

Two Dimensional Lattices
Extending this to a two dimensional $N \times N$ lattice we can see that this lattice consists of N horizontal one dimensional lattices of size N as well as N vertical one dimensional lattices of size N. Combining these we get a total number of possible chains equal to:

$$2 \times N \times N \times \frac{R(R+1)}{2} \tag{5}$$

Therefore the probability of no leakage occuring in an $N \times N$ lattice is:

$$(\frac{2}{3})^{P^3 \times N^2 \times R(R+1)} \tag{6}$$

From this we can again deduce that the probability of leakage during each step is (equation 7):

$$1 - (\frac{2}{3})^{P^3 \times N^2 \times R(R+1)} \tag{7}$$

In any CA it is always the case that $P = 1$ and generally the case that $R = 1$ making the probability of no leakage $(\frac{2}{3})^{2N^2}$. For any reasonable value of N therefore the probability of leakage occuring approaches 1. In an ABM on the other hand $P < 1$. Remembering that $P = \frac{A}{N}$ we can see that in general by inserting $\frac{A}{N}$ for P and simplifying we get:

$$(\frac{2}{3})^{\frac{A^3 R(R+1)}{N}} \tag{8}$$

Based on this we can see that leakage is much more likely to occur in a CA than an ABM. Of course this is based on the assumptions listed at the start of this section. These assumptions are reasonable but there are some final points that we should note:

- The assumption that agents are distributed evenly throughout the lattice is open to question. In an ABM this is not necessarily the case. Often the simulation will have clustering of agents into groups resulting in a higher likelyhood of leakage within these groups and less leakage between groups;

- ABMs require inter agent communication for overall behaviours to emerge. The liklihood that we will design an ABM that has little communications between agents is low. Most simulations will have a large amount of interagent communication thus leading to increased probability of leakage;

- If a *Moore* neighbourhood is used instead of a *von Neumann* neighbourhood then the probability of leakage will increase.

This analysis leads to the conclusion that leakage is a bigger issue in CA than ABM. It also explains the finding in Caron-Lormier et al. (2008) that high population densities enhance the differences in outcomes between asynchronous and synchronous versions of the same ABM.

Any simulation where bounded rationality and/or consistency in the speed at which information spreads through the simulation space are necessary properties should therefore consider using SU to ensure that any results are not due to artefacts of the AU strategy. Of note here are two findings from other researchers.

May (1973) found that delay is a potential contributor to periodicity in systems. AU can interfere with, or even remove, the delay that we would expect in any system with bounded rationality (remember bounded rationality is enforced by the fact that agents can only view their immediate neighbours). Any natural systems where this delay is present would need to be cautious about employing AU or employ SU alongside AU for comparison. Our analysis explains why SU is the preferred updating strategy in traffic flow and pedestrian dynamics simulations as it picks up the important periodic changes in overall state (e.g. traffic jams caused by cars slowing as they pass an accident) by properly modelling the flow of information through the system.

High population densities enhance the differences in outcomes between asynchronous and synchronous versions of the same ABM Caron-Lormier et al. (2008). The fact that in high density populations violations of bounded rationality are more likely to occur gives rise to larger differences in outcomes. This helps explain the stark differences produced by the two versions of the Spatial Iterated Prisoner's Dilemma as the simulation had a maximal population density of one agent per location. The synchronous approach, in contrast, guarantees bounded rationality and imposes a consistent speed on the transmission of information across a simulation space.

CONCLUSIONS

It is commonly assumed that ABM, and particularly ABSS, are more accurate when modelled using AU rather than SU. While asynchronous and synchronous simulations can have different outcomes there is no evidence that the AU based outcomes are more accurate than SU ones. AU posits that by sequentially updating the state of each individual agent while holding all other agents' state constant we obtain the behaviour of an asynchronous system. For this argument to hold each action would have to be instantaneous in time, something that is seldom true in the real world systems.

It has been shown here that the use of asynchronous simulation breaks the bounded rationality principle. It allows agents to be immediately aware of events not within their neighbourhood and to be aware of these events even before other agents who are within the neighbourhood of the events are aware of them. In fact, delayed information is only properly (and consistently) handled by SU.

Our analysis suggests that: Asynchronous updating of a system, by interfering with the speed of transmission of information across the simulation space, will produce less periodicity in the system than synchronous updating.

High density populations within systems are more likely to enhance the differences in outcomes between synchronous and asynchronous implementations of the system. Both of these are borne out in the literature (Caron-Lormier et al. 2008, May 1973).

FURTHER WORK

Recently SU algorithms that can handle the full range of behaviours in ABMs have been developed (Kehoe 2017). This allows SU to be used for complex ABM/ABSS and side by side comparisons to be made between the two approaches to updating (Kehoe 2016). More work needs to be done to see how the choice of updating method effects simulation outcomes and how large (or otherwise) these differences are.

REFERENCES

L. A. Bach*, D. J. Sumpter, J. Alsner, and V. Loeschcke. Spatial evolutionary games of interaction among generic cancer cells. *Journal of Theoretical Medicine*, 5(1):47–58, 2003.

M. Berryman. Review of software platforms for agent based models. Technical report, DTIC Document, 2008.

M. Bezbradica, H. J. Ruskin, and M. Crane. Comparative analysis of asynchronous cellular automata in stochastic pharmaceutical modelling. *Journal of Computational Science*, 5(5):834–840, 2014.

S. Brailsford. DES is alive and kicking. *J Simulation*, 8(1):1–8, Feb 2014. ISSN 1747-7778. URL http://dx.doi.org/10.1057/jos.2013.13.

C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications*, 295(3):507–525, 2001. URL http://EconPapers.repec.org/RePEc:eee:phsmap:v:295:y:2001:i:3:p:507-525.

G. Caron-Lormier, R. W. Humphry, D. A. Bohan, C. Hawes, and P. Thorbek. Asynchronous and synchronous updating in individual-based models. *Ecological Modelling*, 212(34):522 – 527, 2008. ISSN 0304-3800. doi: http://dx.doi.org/10.1016/j.ecolmodel.2007.10.049. URL http://www.sciencedirect.com/science/article/pii/S0304380007006011.

D. Cornforth, D. G. Green, and D. Newth. Ordered asynchronous processes in multi-agent systems. *Physica D: Nonlinear Phenomena*, 204(12):70 – 82, 2005. ISSN 0167-2789. doi: http://dx.doi.org/10.1016/j.physd.2005.04.005. URL http://www.sciencedirect.com/science/article/pii/S0167278905001338.

J. M. Epstein. Agent-based computational models and generative social science. *Complex.*, 4 (5):41–60, May 1999. ISSN 1076-2787. doi: 10.1002/(SICI)1099-0526(199905/06)4:5⟨41::AID-CPLX9⟩3.0.CO;2-F. URL http://dx.doi.org/10.1002/(SICI)1099-0526(199905/06)4:5<41::AID-CPLX9>3.0.CO;2-F.

J. M. Epstein and R. Axtell. *Growing Artificial Societies: Social Science from the Bottom Up*. The Brookings Institution, Washington, DC, USA, 1996. ISBN 0-262-55025-3.

N. Fatès. A guided tour of asynchronous cellular automata. In J. Kari, M. Kutrib, and A. Malcher, editors, *Automata*, volume 8155 of *Lecture Notes in Computer Science*, pages 15–30. Springer, 2013. ISBN 978-3-642-40866-3.

M. Gardner. Mathematical games: The fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, pages 120–123, Oct. 1970.

C. Grilo and L. Correia. Effects of asynchronism on evolutionary games. *Journal of Theoretical Biology*, 269(1):109 – 122, 2011. ISSN 0022-5193. doi: http://dx.doi.org/10.1016/j.jtbi.2010.10.022. URL http://www.sciencedirect.com/science/article/pii/S0022519310005564.

B. A. Huberman and N. S. Glance. Evolutionary games and computer simulations. *Proceedings of the National Academy of Sciences*, 90(16):7716–7718, 1993.

J. Kehoe. Robust reproducibility of agent based models. In *The European Simulation and Modelling Conference*. Inderscience, Oct 2016.

J. Kehoe. *Replication in Agent Based Models using Formal Methods and Multiple Updating Strategies*. PhD thesis, Dublin City University, 2017.

J. Kehoe and J. Morris. A concurrency model for game scripting. In *12th International Conference on Intelligent Games and Simulation*, pages 10–16. EUROSIS, Aug 2011.

S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan. Mason: A multiagent simulation environment. *Simulation*, 81(7):517–527, jul 2005. ISSN 0037-5497. doi: 10.1177/0037549705058073.

R. M. R. M. May. *Stability and complexity in model ecosystems*. Monographs in population biology. Princeton, N.J. Princeton University Press, 1973. ISBN 0-691-08125-5.

M. J. North, N. T. Collier, J. Ozik, E. R. Tatara, C. M. Macal, M. Bragen, and P. Sydelko. Complex adaptive systems modeling with repast simphony. *Complex Adaptive Systems Modeling*, 1(1):1–26, 2013.

M. A. Nowak and R. M. May. Evolutionary games and spatial chaos. *Nature*, 359(6398):826–829, Oct. 1992. doi: 10.1038/359826a0.

B. S. Onggo. Running agent-based models on a discrete-event simulator. In *Proceedings of the 24th European Simulation and Modelling Conference*, pages 25–27. Citeseer, 2010.

S. Railsback, S. Lytinen, and V. Grimm. Stupidmodel and extensions: A template and teaching tool for agent-based modeling platforms, 2005.

G. D. Ruxton and L. A. Saravia. The need for biological realism in the updating of cellular automata models. *Ecological Modelling*, 107(23):105 – 112, 1998. ISSN 0304-3800. doi: http://dx.doi.org/10.1016/S0304-3800(97)00179-8.

B. Schönfisch and A. de Roos. Synchronous and asynchronous updating in cellular automata. *Biosystems*, 51(3):123 – 143, 1999. ISSN 0303-2647. doi: http://dx.doi.org/10.1016/S0303-2647(99)00025-8. URL http://www.sciencedirect.com/science/article/pii/S0303264799000258.

P.-O. Siebers, C. M. Macal, J. Garnett, D. Buxton, and M. Pidd. Discrete-event simulation is dead, long live agent-based simulation! *J. Simulation*, 4(3):204–210, 2010.

SIMULATION DRIVEN ENGINEERING

MODEL-BASED SIMULATION OF A HYDRAULIC OPEN-LOOP ROTARY TRANSMISSION WITH AUTOMATIC REGULATION OF HYDRAULIC MOTOR (PART 1: MODELLING)

Mait Harf
Institute of Software Science
Tallinn University of Technology
Akadeemia tee 21, 12618 Tallinn, Estonia
E-mail: mait@cs.ioc.ee

Gunnar Grossschmidt
Department of Mechanical and Industrial Engineering
Tallinn University of Technology
Ehitajate tee 5, 19086 Tallinn, Estonia
E-mail: gunnar.grossschmidt@ttu.ee

KEYWORDS
Hydraulic rotary transmission, automatic regulation of hydraulic motor, multi-pole mathematical model.

ABSTRACT

Model-based simulation of a hydraulic open-loop rotary transmission with automatic regulation of hydraulic motor is considered in the paper. The approach is based on multi-pole modelling and intelligent simulation. In Part 1 of the paper the functional scheme of the transmission is described and corresponding multi-pole model is introduced. Mathematical multi-pole models of functional elements for stationary conditions and for dynamic transient responses are presented. In Part 2 of the paper simulation of steady–state conditions and dynamic responses of the hydraulic transmission is considered. A high-level graphical environment CoCoVila (compiler compiler for visual languages) is used as a tool for describing models and performing simulations. Object-oriented multi-pole models, visual programming environment, automatic program synthesis and distributed computing are as original approach in simulation of fluid power systems.

1 INTRODUCTION

Hydraulic rotary transmissions offer fast response, precise speed control from zero to maximum under varying loads, fast and stepless adjustment of speed, torque and power plus smooth and controllable acceleration, ability to be stalled without damage and easy controllability. Open-loop circuits are generally used for hydraulic transmissions in stationary applications.
Hydraulic open-loop rotary transmission under conside-ration consists of a variable displacement pump driven by a diesel engine and a variable displacement hydraulic motor (Murrenhoff 2005, Parambath 2016). The hydraulic motor is automatically regulated depending on load moment of the transmission.
Designing of such hydraulic systems in experimental way is time consuming and expensive. Using computer simulation enables to consider different configurations of systems and choose optimal parameters.

2 FUNCTIONAL SCHEME OF AN OPEN-LOOP HYDRAULIC ROTARY TRANSMISSION

The functional scheme of an open-loop hydraulic rotary trans- mission with automatically regulated hydraulic motor is shown in Fig.1.

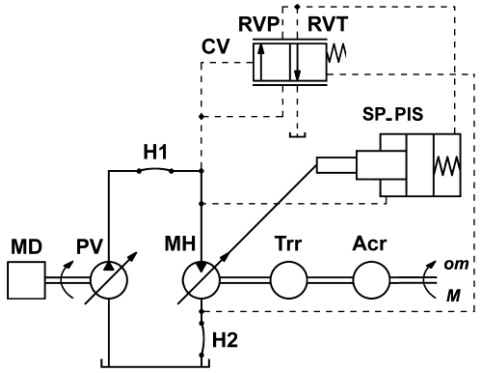


Figure 1: The functional scheme of an open-loop hydraulic rotary transmission

In Fig. 1 components of transmission are denoted: **MD** – diesel engine, **PV** – variable displacement hydraulic pump, **MH** – variable displacement hydraulic motor, **H1, H2** – hydraulic hoses, **CV** – control valve, **RVP, RVT** – control valve inlet and outlet throttle edges, **SP_PIS** – hydraulic motor swash plate with regulating piston, **Trr** – speed reducer, **Acr** – rotation actuator.

In order to maximize power usage it is reasonable to keep the transmission in operation at constant power. To achieve this, changing the load moment of the transmission requires the hydraulic motor MH to change rotation speed. Rotation speed of hydraulic motor is regulated by the swash plate. The swash plate position angle is automatically regulated by the hydraulic servo-system, containing an asymmetric piston in a cylinder SP_PIS and a servo-valve CV with throttle edges RVP and RVT. Feedback to the servo-valve CV is applied by different pressures at the hydraulic motor characterizing the load moment of the transmission.

3 MULTI-POLE MODEL OF THE TRANSMISSION

Multi-pole mathematical models (Grossschmidt and Harf 2009 Part 1, Grossschmidt and Harf 2010, Grossschmidt and Harf 2012) are used that adequately describe the physical processes in hydraulic and mechanical systems. Both direct actions and feedbacks are expressed in component models.
Using multi-pole models allows to describe the required complexity for each component. A component model can enclose nonlinear equations, inner iterations, logic functions and calculation programs as relations.
The causalities of multi-pole models will be selected depending on:

- process of simulation (static, steady-state, transient response)
- the physical causality of given input and output variables
- table of multi-pole models of components and sub-systems being used
- the obligatory causalities of separate dependencies and devices
- possible variants of connecting the multi-pole models
- wish to avoid the "mathematical stiff" dependencies
- multi-pole models of typical subsystems and systems.

Model of steady-state conditions of the hydraulic rotary transmission is shown in Fig. 2.

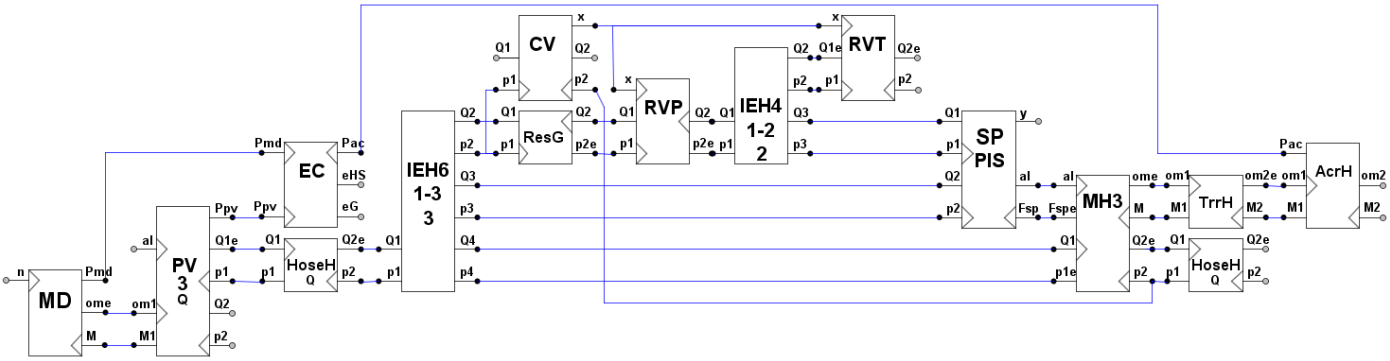


Figure 2: Model of steady-state conditions of an open-loop hydraulic rotary transmission

In Fig. 2 components of transmission are denoted thus: **MD** – diesel engine, **PV3_Q** – variable displacement hydraulic pump, **MH3** – variable displacement hydraulic motor, **HoseH_Q** – hydraulic hoses, **CV** – control valve, **RVP, RVT** – control valve inlet and outlet throttle edges, **SP_PIS** – hydraulic motor swash plate with regulating piston, **TrrH** – speed reducer, **AcrH** – rotation actuator, **EC** – efficiency coefficient calculator, **IEH** – hydraulic interface elements, **ResG** – hydraulic resistor.

The dynamics model of the rotary transmission contains additional components hydraulic resistors and a clutch (see the dynamic simulation task description in Part 2).

4. MATHEMATICAL MULTI-POLE MODELS OF COMPONENTS

4.1 Diesel Engine MD

Inputs: n – rotation speed, rpm, M – load moment, Nm.
Outputs: ω – rotation speed, rad/s, Pmd – power used by diesel engine, W.

Model of Steady-state Conditions

Rotation speed ω of diesel engine, rad/s

$$\omega = \omega 0 - k * M / Mmax, \quad (1)$$

where
rotation speed at load moment M = 0

$$\omega 0 = 2 * \pi * n0 / 60, \quad (2)$$

k – coefficient of the static characteristic, rad/s
n0 – rotation speed at load moment M = 0, rpm
Mmax – maximum moment of diesel engine, Nm.

The maximum moment Mmax of the diesel engine depends on its rotation speed (Mollenhauer and Tschöke 2010). Used dependence is shown in Fig. 3.

Moment, developed by diesel engine, Nm

$$Md = M / \eta dm + M0, \quad (3)$$

where
ηdm – diesel engine efficiency coefficient
M0 – resisting moment at idle run, Nm.

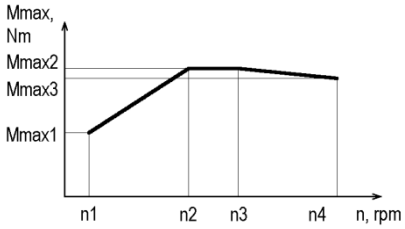


Figure 3: Maximum moment Mmax of diesel engine

Power Pmd available from diesel engine, W

$$Pmd = Md * \omega. \quad (4)$$

Model of Dynamics

Difference of moment, developed by diesel engine, Nm

$$dMd = (\Delta/T) * ((Mmax/k) * (\omega 0 - \omega) - Md), \quad (5)$$

where
Δ – simulation time step, s
T – time constant, s.

Difference of diesel engine rotation speed, rad/s

$$dom = (\Delta / J)*(Md - M), \quad (6)$$

where
J – inertia moment of diesel engine, kgm^2/rad.

Values of Parameters

The values of the parameters of the functional elements (components) used in the simulations must be specified. Some parameters, are specified by the requirements of the transmission under consideration and are taken from the catalogues of producing companies. Usually catalogue data do not contain all the information required for simulation. Some components can be original for the transmission, but values of their parameters cannot be found from literature. Missing values of parameters must be initially chosen approximately and are to be adjusted during the simulations. Further in this chapter final values of parameters of all the components used in the simulations e.g. adjusted during the simulations are presented.

Values of parameters of the diesel engine are as follows.

k = 5, n1 = 1000 rpm, n2 = 1400 rpm, n3 = 1600 rpm, n4 = 2000 rpm, Mmax1 = 130 Nm, Mmax2 = 260 Nm, Mmax3 = 240 Nm, ηdm = 0.95, M0 = 0.1 Nm, T = 0.01 s, J = 0.15 kgm^2 /rad.

4.2 Variable Displacement Pump PV3_Q

Inputs: α – position angle of the pump swash plate, deg, ω1 – rotation speed, rad /s, p1 – pressure in pump outlet, Pa, p2 – pressure in pump inlet, Pa.
Outputs: Q1 – outlet volumetric flow, m^3/s, Q2 – inlet volumetric flow, m^3/s, M1 – reaction moment of pump, Nm, Ppv – power for driving the pump, W.

Model of Steady-state Conditions

Working volume V1r of pump, m^3/rad

$$V1r = V1max * (\tan(\alpha * \pi / 180) / \tan(\alpha max * \pi / 180)) / (2 * \pi), \quad (7)$$

where
V1max – maximum pump working volume, m^3/rev
αmax – maximum position angle of the pump swash plate, deg.

Volumetric efficiency coefficient of the pump

$$\eta vol = 1 - kvol * (p1 - p2), \quad (8)$$

where
kvol1 – coefficient of dependence of ηvol1 on pressure drop (p1 – p2), 1 / Pa
p1 – pressure in pump outlet, Pa
p2 – pressure in pump inlet, Pa.

Volumetric flow Q1 in pump outlet, m^3/s

$$Q1 = \omega 1 * V1r * \eta vol, \quad (9)$$

where
ω1 – rotation speed of pump, rad / s.

Hydro-mechanical efficiency coefficient

$$\eta hm = 1 - khm * (p1 - p2), \quad (10)$$

where
khm – coefficient of dependence of ηhm on pressure drop (p1 – p2), 1/Pa.

Output moment

$$M1 = V1r * (p1 - p2) / \eta hm + kf * (p1 - p2) + M0, \quad (11)$$

where
kf – coefficient of dependence of friction moment on pressure drop (p1 – p2), m^3
M0 – resisting moment at idle run, Nm.

The model is used for both steady-state conditions and dynamics considering that the moment of inertia of the pump rotor is added to the clutch.

Values of Parameters

V1max = 40e–6 m^3/rev (6.366 m^3/rad), αmax = 35 deg (0.611 rad), kvol = 2e–9 1/Pa, khm = 4e–9 1/Pa, kf = 1e–8 m^3, M0 = 0.1 Nm.

4.3 Variable Displacement Hydraulic Motor MH3

Inputs: α – position angle of the motor swash plate, deg, M – load moment, Nm, p2 – outlet pressure, Pa.
Outputs: p1 – inlet pressure, Pa, Q2 – volumetric flow, m^3/s, Fsp – reaction force of swash plate, N, ω – rotation speed, rad/s.

Working volume Vr of hydraulic motor, m^3/rad

$$Vr = Vmax *(\tan(\alpha * \pi / 180) / \tan(\alpha max * \pi /180)) / (2 * \pi), \quad (12)$$

where
Vmax – maximum hydraulic motor working volume, m^3/rev
αmax – maximum position angle of the hydraulic motor swash plate, deg.

Output pressure, Pa

$$p1 = (M / (Vr * \eta hm) + M0 /Vr + p2), \quad (13)$$

where
ηhm – hydro-mechanical efficiency coefficient

$$\eta hm = 1 - khm * (p1 - p2) \quad (14)$$

khm – coefficient of dependence of ηhm on pressure drop (p1 – p2), 1/Pa
M0 – resisting moment at idle run, Nm.

Volumetric efficiency coefficient

$$\eta vol = 1 - kvol * (p1 - p2), \quad (15)$$

where
kvol – coefficient of dependence of ηvol on pressure drop (p1 – p2), 1 / Pa.

Output volumetric flow, m^3/s

$$Q2 = Q1 * \eta vol, \quad (16)$$

where
Q1 – input volumetric flow, m^3/s.

Output rotation speed ω, rad/s

$$\omega = Q2 / Vr. \quad (17)$$

Reaction force of swash plate, N

$$Fsp = ksp * (p1 - p2) * A * n / 2, \quad (18)$$

where
ksp – coefficient of proportionality on pressure drop (p1–p2), m^2
A – piston area, m^2 ($A = \pi * d^2 / 4$)
d – diameter of piston, m
n – number of pistons.

The model is used for both steady–state conditions and dynamics considering that the moment of inertia of pump rotor is added to speed reducer.

Values of Parameters

Vmax = 55e–6 m^3/rev (8.7535e–6 m^3/rad), αmax = 35 deg (0.611 rad), kvol = 3e–9 1/Pa, khm = 2e–9 1/Pa, M0 = 0.1 Nm, ksp = 0.005, d = 0.01 m, n = 7.

4.4 Control Valve CV with Throttle Edges RVP and RVT

A section view of the control valve CV with throttle edges RVP and RVT is shown in Fig.4.

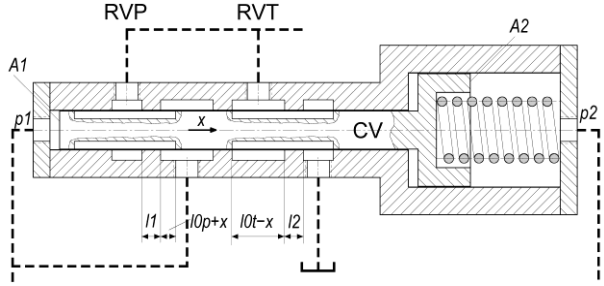


Figure 4: Section view of the control valve CV with throttle edges RVP and RVT

Pressures p1 and p2 at the ends of CV are inlet and outlet pressures of hydraulic motor MH3. Two meter-in throttle edges RVP are of length l1+l0p+x, two meter-out throttle edges RVT are of length l2+l0t–x.

4.4.1 Control Valve CV

Inputs: p1 – pressure in hydraulic motor inlet, Pa,
p2 – pressure in hydraulic motor outlet, Pa.
Output: x – displacement of control valve, m.

Model of Steady-state Conditions

Force acting to control valve, m

$$Fx = p1*A1 – p2*A2 – Ffr – F0, \quad (19)$$

where
A1, A2 – effective areas of valve, m^2

$$A1 = \pi * d1^2 / 4, \quad A2 = \pi * d2^2 / 4$$

d1, d2 – diameters of effective areas of valve, m
Ffr – friction force, N

$$Ffr = (Ffr0 + kfr * (p1 – p2)) * sign(v), \quad (20)$$

where
Ffr0 – initial friction fore, N
kfr – coefficient of dependence of Ffr on pressure drop (p1 – p2), m^2
F0 – force of spring preliminary deformation, N

$$F0 = x0 * cx, \quad (21)$$

x0 – length of spring preliminary deformation, m
cx – control valve spring stiffness, N/m

$$cx = G * d^4 / (8 * n * D^3), \quad (22)$$

where
G – shear modulus, N/m^2
d – diameter of spring wire, m
D – diameter of spring, m
n – number of turns of the spring.

Displacement of control valve

$$x = Fx / cx. \quad (23)$$

Model of Dynamics

Difference of valve velocity, m/s

$$dv = (\Delta /m) * (Fx – x * cx – h * v – Ffr – F0), \quad (24)$$

where
Δ – simulation time step, s
m – sum of valve and 1/3 of spring mass, kg
h – damping coefficient, Ns/m,
v – velocity of control valve, m/s
Ffr – friction force, N

Ffr = (Ffr0 + kfr * (p1 – p2) – v * 0.7*Ffr0 / 0.02) * signv,
if v = 0...0.02 m/s,
Ffr = 0.3 *(Ffr0 + kfr * (p1 – p2) * signv,
if v > 0.02 m/s. (25)

Difference of valve displacement, m

$$dx = \Delta * v. \quad (26)$$

Output volumetric flows, m^3/s

$$Q1 = A1 * v, \quad (27)$$
$$Q2 = A2 * v.$$

Values of Parameters

d1 = 0.008 m, d2 = 0.02 m, Ffr0 = 0.1 N, kfr = 1e–8 m^2, G = 8e11 N/m^2, d =0.0018 m, D = 0.011 m, n = 8, mv = 0.3 kg, ms = 0,06 kg, h = 50 Ns/m, x0 = 0.0005 m.

4.4.2 Control Valve Meter-in Throttle Edges RVP

Inputs: x – displacement of control valve, m,
p1 – inlet pressure, Pa, Q2 – outlet volumetric flow, m^3/s.
Outputs: Q1 – inlet volumetric flow, m^3/s, p2 – outlet pressure, Pa.

Output pressure (Murrenhoff 2005)

$$p2 = p1 – 2 * (12 * \nu * \rho * (l0p + l1 + x) / (8 * b * R^3)) * Q2, \quad (28)$$

where
ν – kinematic viscosity of fluid, m^2/s
ρ – density of fluid, kg/m^3
l0p – initial position of throttle slot, m
l1 – length of separation wall, m
b – depth of throttle slot, m
R – radius of end of throttle slot (width of edge h = 2 * R).

Output volumetric flow, m^3/s

$$Q1 = Q2. \quad (29)$$

Values of Parameters

l0p = 0.0005 m, l1 = 0.005 m, b = 0.001 m, R = 0.00014 m.

4.4.3 Control Valve Meter-out Throttle Edges RVT

Inputs: x – displacement of control valve, m, p1 – inlet pressure, Pa, p2 – outlet pressure, Pa.

Outputs: Q1 – inlet volumetric flow, m^3/s,
Q2 – outlet volumetric flow, m^3/s.

Output volumetric flows (Murrenhoff 2005)

$$Q1 = ((8 * b * R^3) / (2 * 12 * \nu * \rho * (l0t + l2 – x))) * (p1 – p2), \quad (30)$$

$$Q2 = Q1, \quad (31)$$

where
b – depth of throttle slot, m
l0t – initial position of the throttle slot, m
l2 – length of separation wall, m.

Values of Parameters

l0t = 0.013 m, l2 = 0.005 m, b = 0.001 m, R = 0.00012 m.

4.5 Hydraulic Motor Swash Plate with Regulating Piston SP_PIS

Inputs: p1 – inlet pressure acting to piston area A1, Pa,
p2 – inlet pressure acting to piston area A2, Pa,
Fsp – reaction force of swash plate, N.
Outputs: α – position angle of the hydraulic motor swash plate, deg, Q1, Q2 – volumetric flows, m^3/s, y – regulating piston displacement, m.

Model of Steady-state Conditions

Force acting to spring, N

$$Fy = p2 * A2 – p1 * A1 – Fsp – Ffr – F0, \quad (32)$$

where
A1, A2 – piston effective areas, m^2
(A1 = π * dcy^2 / 4, A2 = A1 – π * dro^2 / 4),
dcy – diameter of regulating cylinder, m
dro – diameter of regulating piston rod, m
Fsp – reaction force of swash plate, N

$$Fsp = ksp * (p1 - p2) * A * n/2 \quad (33)$$

ksp – coefficient of proportionality on pressure drop (p1–p2), m^2

A – pump piston working area, m^2

$$A = \pi * d^2 / 4$$

d – hydraulic motor piston diameter, m
n – number of hydraulic motor pistons
Ffr – friction force, N, see formula (20)
F0 – force of spring preliminary deformation, N

$$F0 = y0 / cy,$$

y0 – length of spring preliminary deformation, m
Displacement of regulating piston, m

$$y = Fy / cy, \quad (34)$$

where
Fy – Force acting to spring, N, see formula (32)
cy – stiffness of regulating piston spring, N/m , see formula (22).

Position angle α of the hydraulic motor swash plate, deg

$$\alpha = \arctan (y / ry) * (180 / \pi), \quad (35)$$
$$\text{if } (\alpha <= \alpha min) \ \alpha = \alpha min,$$
$$\text{if } (\alpha >= \alpha max) \ \alpha = \alpha max,$$

where
ry – distance between regulating piston rod and hydraulic motor axis, m
αmin, αmax – minimum and maximum position angle of the hydraulic motor swash plate, deg.

Model of Dynamics

Difference of piston velocity, m/s

$$dv = (\Delta / mvp) * (Fy – y * cy – Ffr – h * v – F0), \quad (36)$$

where
Δ – simulation time step, s
mvp – sum of piston, swash plate and 1/3 spring mass, kg

$$mvp = mv + msp + ms/3$$

y – displacement of regulating piston, m
cy – stiffness of regulating piston spring, N/m, see formula (22)
Ffr – friction force, N, see formula (25)
h – damping coefficient, Ns/m
v – velocity of regulating piston, m / s
F0 – force of spring preliminary deformation, N

$$F0 = y0 * cy,$$

y0 – length of spring preliminary deformation, m.
Difference of regulating piston displacement, m

$$dy = \Delta * v. \quad (37)$$

Volumetric flows, m^3/s

$$Q1 = – A1 * v, \quad (38)$$
$$Q2 = A2 * v.$$

Values of Parameters

dcy = 0.035 m, dro = 0.0247 m, ry = 0.09 m, Ffr0 = 1 N, kfr = 1e–8 m^2, G = 8e11 N/m^2, d = 0.0020 m, D = 0.011 m, n = 21, αmin = 10 deg, αmax = 35 deg, mv = 0.3 kg, msp = 0.2 kg, ms = 0.06 kg, h = 50 Ns/m, y0 = 0.003 m.

4.6 Speed Reducer TrrH

Inputs: ω1– inlet rotation speed, rad/s,
M2 – load moment, Nm.
Outputs: ω2– outlet rotation speed, rad/s,
M1 – inlet moment, Nm.

Model of Steady-state Conditions

Output moment for stationary movement, Nm

$$M1 = M2 * i / \eta m + M10, \quad (39)$$

where
M2 – input load moment, Nm
i – transfer factor
ηm – mechanical efficiency coefficient
M10 – resisting moment at idle run, Nm.

Output rotation speed for steady-state conditions, rad/s

$$\omega 2 = \omega 1 * i. \quad (40)$$

Model of Dynamics

Difference of output moment, Nm

$$dM1 = (\Delta / e) * (\omega 1 * i – \omega 2 – hM * (M1 – M2 * i / \eta m – M10)), \quad (41)$$

where
Δ – simulation time step, s
e– torsion elasticity, rad / Nm
hM – damping constant according to difference of moments (M1 – M2), rad / Nms.

Difference of output rotation speed, rad/s

$$d\omega 2 = (\Delta / J) * (M1 – M2 – hom * (\omega 1 – \omega 2)), \quad (42)$$

where
J – inertia moment, kgm^2/rad
hom – damping constant according to angular velocity difference (ω1 – ω2), Nms/rad.

Values of Parameters

i = 0.5, ηm = 0.95, M10 = 0.1 Nm, e = 3e-5 rad /Nm, hM = 0.1 rad/Nms, J = 0.002 kgm^2/rad, hom = 0.001 Nms/rad.

4.7 Rotary Actuator AcrH

Inputs: ω1 – inlet rotation speed, rad/s,
M2 – outlet moment, Nm.
Outputs: ω2 – outlet rotation speed, rad/s, M1 – inlet moment, Nm.

Model of Steady-state Conditions

Output moment, Nm

$$M1 = M2 / \eta m + M10, \quad (43)$$

where
ηm – mechanical efficiency coefficient
M10 – resisting moment at idle run, Nm.

Output rotation speed, rsd/s

$$\omega 2 = \omega 1. \quad (44)$$

Output power Pac, W

$$Pac = M2 * \omega 2. \quad (45)$$

Model of Dynamics

Difference of output moment, Nm

$$dM1 = (\Delta/e) * (\omega 1 – \omega 2 – hM * (M1 – M2 – M10)), \quad (46)$$

where
Δ – simulation time step, s
e – torsion elasticity, rad / Nm
hM – damping constant according to moment difference (M1 – M2), rad / Nms
M10 – resisting moment at idle run, Nm.

Difference of output rotation speed, rad/s

$$dom2 = (\Delta / J) * (M1 - M2 / \eta m - M10 + hom * (om1 - om2)), \quad (47)$$

where

Δ – simulation time step, s
J – inertia moment, kgm^2/rad
ηm – mechanical efficiency coefficient
M10 – resisting moment at idle run, Nm
hom – damping constant according to difference of angular velocities ($\omega 1 - \omega 2$), Nms/rad.

Values of Parameters

ηm = 0.95, M10 = 1 Nm, e = 3e–5 rad /Nm, hM = 0.1 rad/Nms, J = 0.05 kgm^2/rad, hom = 0.001Nms/rad.

4.8 Hydraulic Interface Elements

Hydraulic interface elements express:

- equality of output pressures, if pressure of one pole is given as input
- the equation of continuity of volumetric flows:

$$Q1 = Q2 + Q3 + Q4 \quad (IEH6\_1–3), \quad (48)$$

$$Q1 = Q2 + Q3 \quad (IEH4\_1–2). \quad (49)$$

4.9 Efficiency Coefficient Calculator EC

Inputs: Pmd – power used from diesel engine, W, Ppv – outlet power of the pump, W, Pac – outlet power of actuator, W.

Outputs: eHS – efficiency coefficient of the hydraulic rotary transmission without pump, clutch and engine, eG – efficiency coefficient of the hydraulic rotary transmission with pump, clutch and engine.

Efficiency coefficients

$$eHS = Pac / Ppv, \quad (50)$$

$$eG = Pac / Pmd. \quad (51)$$

4.10 Other Models

Models of physical properties of fluids and models of hydraulic resistors are described in (Grossschmidt and Harf 2009a).

Model CJh of a clutch with rotor of the pump is described in (Grossschmidt and Harf 2009b).

Model of a hydraulic hose HoseH_Q is presented in (Grossschmidt and Harf 2017). Values of main parameters of hoses: inner diameter d = 0.02 m, length l = 2 m.

SUMMARY

In the paper multi-pole modelling of a hydraulic open-loop rotary transmission with automatic regulation of hydraulic motor has been considered.

The multi-pole mathematical models of components having various causalities are introduced for composing a model of the whole system. The mathematical models are different for steady-state conditions and for dynamic responses.

The multi-pole mathematical models of components: diesel engine, variable displacement hydraulic pump, variable displacement hydraulic motor, components of hydraulic motor regulating system, speed reducer and rotary actuator are described.

The multi-pole model of the transmission is as the basis to describe the visual simulation tasks and perform the simulation itself.

The visual programming environment and simulation results for both steady-state conditions and dynamic transient responses are presented and discussed in Part 2 of the paper.

ACKNOWLEDGEMENTS

This research was supported by the Estonian Ministry of Research and Education institutional research grant no. IUT33-13, the Innovative Manufacturing Engineering Systems Competence Centre IMECC and Enterprise Estonia (EAS) and European Union Regional Development Fund (project EU48685).

REFERENCES

Grossschmidt, G. and M. Harf 2009a. "COCO-SIM – Object-oriented Multi-pole Modeling and Simulation Environment for Fluid Power Systems, Part 1: Fundamentals". *International Journal of Fluid Power*, 10(2), 2009, 91–100.

Grossschmidt, G. and M. Harf. 2009b. "COCO-SIM – Object-oriented multi-pole modelling and simulation environment for fluid power systems. Part 2: Modelling and simulation of hydraulic-mechanical load-sensing system*." International Journal of Fluid Power*,10(3),71 - 85.

Grossschmidt, G. and M. Harf 2010. "Simulation of hydraulic circuits in an intelligent programming environment, Part 1". *7th International DAAAM Baltic Conference "Industrial engineering"*, 22-24 April 2010, Tallinn, Estonia, 148 -161.

Grossschmidt, G. and M. Harf 2012. "Multi-pole modeling and intelligent simulation of technical chain systems, Part 1." *In: Proceedings of the 8th International Conference of DAAAM Baltic "Industrial Engineering", [vol 2];* 19-21st April 2012, Tallinn, Estonia; (Red.) Otto, Tauno, Tallinn; Tallinn University of Technology, 2012, 458-471.

Grossschmidt, G. and M. Harf 2017. "Model-based simulation of hydraulic hoses in an intelligent environment." *International Journal of Fluid Power,* Vol. 19, No. 1, March 2018, Taylor & Francis, 27 - 41.

Harf, M. and G. Grossschmidt 2014. "Multi-pole Modeling and Intelligent Simulation of Control Valves of Fluid Power Systems, Part 1." *In: Proceedings of the 9th International Conference of DAAAM Baltic "Industrial Engineering"*; 24-26 April 2014, Tallinn, Estonia; (Red.) Otto, Tauno, Tallinn; Tallinn University of Technology, 2014, 17-28.

Mollenhauer, K. and H. Tschöke 2010. "Handbook of Diesel Engines". *Springer-Verlag*, Berlin Heidelberg, 2010.

Murrenhoff, H. 2005. "Grundlagen der Fluidtechnik, Teil 1: Hydraulik", 4. neu überarbeitete Auflage. *Institut für fluidtechnische Antriebe und Steuerungen*, Aachen, 2005.

Parambath, J. 2016. "Industrial Hydraulic Systems – Theory and Practice". *Universal–Publishers Boca Raton*, Florida, USA, 2016.

MODEL-BASED SIMULATION OF A HYDRAULIC OPEN-LOOP ROTARY TRANSMISSION WITH AUTOMATIC REGULATION OF HYDRAULIC MOTOR (PART 2: SIMULATION)

Mait Harf
Institute of Software Science
Tallinn University of Technology
Akadeemia tee 21, 12618 Tallinn, Estonia
E-mail: mait@cs.ioc.ee

Gunnar Grossschmidt
Department of Mechanical and Industrial Engineering
Tallinn University of Technology
Ehitajate tee 5, 19086 Tallinn, Estonia
E-mail: gunnar.grossschmidt@ttu.ee

KEYWORDS
Hydraulic rotary transmission, automatic regulation of hydraulic motor, intelligent simulation environment, visual task description, automatic program synthesis.

ABSTRACT

Model-based simulation of a hydraulic open-loop rotary transmission with automatic regulation of a hydraulic motor is considered in the paper. The approach is based on multi-pole modelling and intelligent simulation. In Part 1 of the paper, the functional scheme of the transmission is described and the corresponding multi-pole model is introduced. Mathematical multi-pole models of functional elements for stationary conditions and for dynamic transient responses are presented. In Part 2 of the paper simulation of steady–state conditions and dynamic responses of the hydraulic transmission is considered. Visual programming environ-ment CoCoViLa is used as a tool for modelling and simulation. CoCoViLa supports declarative programming in a high-level language and automatic program synthesis. Using encapsulation in multi-pole models of components enables to avoid solving large differential equation systems when performing simulations. Simulation results are presented and discussed. The proposed technology enables to find optimal solutions in design of hydraulic open-loop rotary transmission.

1 INTRODUCTION

In available literature rotary transmissions containing only a pump and a hydraulic motor are observed. Mainly output rotation frequency, moment, power and efficiency coefficient are calculated (Murrenhoff 2005, Parambath 2016).
In the present paper simulation results of the open-loop rotary transmission using CoCoViLa environment (Grigorenko et al 2005, Grigorenko and Tyugu 2006, Kotkas et al 2011) are presented. The simulations are performed on multi-pole mathematical models, described in Part 1 of the paper.
Automatic regulation of a hydraulic motor in a rotary transmission is based on using feedback from load applied.
Experimental setting up parameters of such systems is time consuming and expensive. Using computer simulation enables to find optimal parameters faster and cheaper.
The approach based on using model-based simulation and the CoCoViLa environment have been used for modelling and simulation of several complex fluid power systems (Grossschmidt and Harf 2009, 2012, 2016, Harf and Grossschmidt 2013, 2015).

2 SIMULATION ENVIRONMENT

CoCoViLa is a tool for model-based software development with a visual language support. It uses automatic synthesis of programs for translating declarative specifications of simulation problems into executable code.
Using a visual simulation environment like CoCoViLa enables us to describe multi-pole models graphically which facilitates the model development. Automatic synthesis of the calculation algorithms allows focusing on designing models of fluid power systems instead of constructing and solving simulation algorithms.
CoCoViLa is developed as an open-source software, its extensions can be written in Java and included into simulation packages. CoCoViLa is implemented in the Institute of Cybernetics at the Tallinn University of Technology. The CoCoViLa environment is platform-independent and free.

3 SIMULATION PROCESS ORGANIZATION

Using visual specifications of described multi-pole models of fluid power system components one can compose models of various fluid power systems for simulating statics or steady-state conditions and dynamic responses.
When simulating statics or steady-state conditions fluid power system behaviour is simulated depending on different values of input variables. Number of calculation points must be specified.
When simulating dynamic behaviour, transient responses in certain points of the fluid power system caused by applied disturbances are calculated. Disturbances are considered as changes of input variables of the fluid power system (pressures, volumetric flows, load forces or moments, control signals, etc.). Typically disturbances of step, pulse or sine form are used. For integrations in dynamic calculations the fourth-order classical Runge-Kutta method is used in component models.
Static, steady-state and dynamic computing processes are organized by the corresponding process managers (static Process, dynamic Process). To follow the system behaviour, the concept of state is invoked. State variables are introduced for a component in order to characterize its behaviour at the current simulation step.
A simulation task requires sequential computing states until some satisfying final state is reached. A final state can be computed from a given initial state if a function exists that calculates the next state from known previous states. This

function is to be constructed automatically by the CoCoViLa program synthesizer.

When calculating next state from previous states loop dependencies between poles of component models may occur. A special technique is used for calculating variables in loop dependences that may appear when multi-pole models of components are connected together. One variable in each loop is split and iteratively recomputed to find its value satisfying the loop dependency.

State variables and split variables must be described in component models. When building a particular simulation task model and performing simulations, state variables and split variables are used automatically if needed.

When building up a simulation task scheme all the parameters of components must be provided with values.

Initial values of state variables and variables requiring iterations characterize the model in the beginning of the simulation. Specifying precise initial values is not critical for statics and steady-state conditions. Approximate initial values must be set. For dynamics setting initial conditions is critical to guarantee reliable transient responses calculated. Trustful initial conditions can be obtained only as a result of iterative calculations of transient responses in the case of zero disturbances which must be performed as a separate simulation.

Dynamic simulation time step must be chosen to be short enough in order to calculate transient responses of higher frequencies and rapid transitions. In the simulation examples concerning system under discussion time step Δt = 4e-6 s is used.

Maximum number of iterations, adjusting factor for iterations, allowed absolute and relative errors are to be specified for calculating variables in loops.

Principles of simulation process organization have been considered in more details in (Grossschmidt and Harf 2016).

Physical properties of working fluid (density ρ, kinematic viscosity ν and compressibility factor β) are calculated at each simulation step depending on the average of input and output pressure in the component. In all the simulations below the hydraulic fluid HLP46 is used. The initial values of physical properties of fluid HLP46 at zero pressure and at temperature 40°C are: density ρ = 873 kg/m^3, viscosity ν = 46e-6 m^2/s, compressibility factor β = 6.1e-10 1/Pa, air relative content in fluid vol = 0.08.

4 SIMULATION

4.1 Requirements

The hydraulic transmission under consideration is a complicated fluid power system. There are a number of requirements to operating parameters the simulated system must meet. The main requirements are:

- actuator maximum rotation speed ωmax = 200 rad/s
- actuator maximum load moment Mmax = 450 Nm
- hydraulic motor maximum inlet pressure pmax = 3e7 Pa

The additional requirements for dynamic behaviour must be followed:

- actuator inertia moment J = 0.01...0.1 kgm^2/rad
- maximum damping time of oscillations t = 1.5 s
- maximum duration of dynamic process t = 2 s

Behaviour of the system is influenced by great number of constructive parameters such as:

- parameters of diesel engine and hydraulic pump
- parameters of hydraulic motor and speed reducer
- geometry of throttle edges
- parameters of springs (diameters of spring and wire, shear modulus, number of turns)
- construction parameters of regulating piston and cylinder,
- dimensions of hydraulic resistors
- models and values of friction forces
- working areas of regulating valves, etc.

After building up the model, the approximate initial values of parameters have been set up. A number of simulations have been performed to configure the system to function and adjust parameters. Adjusted values of parameters are presented in Part 1 of the paper. In the following chapters final results of simulations are presented.

4.2 Simulation of Steady-state Conditions

Simulation task of steady-state conditions hydraulic open-loop rotary transmission with automatic regulation of a hydraulic motor is shown in Fig. 1.

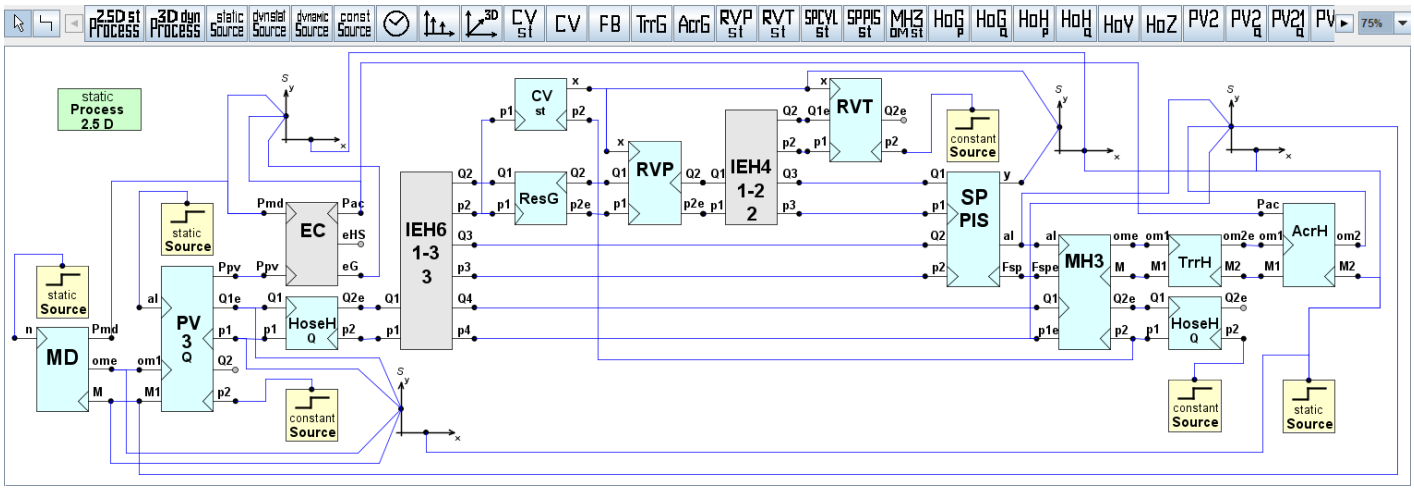


Figure 1: Steady-state condition simulation task of open-loop rotary transmission

In Fig. 1 components of transmission are denoted: **MD** – diesel engine, **PV3_Q** – variable displacement hydraulic pump, **MH3** – variable displacement hydraulic motor, **HoseH_Q** – hydraulic hoses, **CV_st** – control valve, **RVP, RVT** – control valve inlet and outlet throttle edges, **SP_PIS** – hydraulic motor swash plate with regulating piston, **TrrH**

– speed reducer, **AcrH** – rotation actuator, **EC** – efficiency coefficient calculator, **IEH** – hydraulic interface elements, **ResG** – hydraulic resistor, **static Process** – process manager.

In Fig. 2...6 results of simulations of steady-state conditions are shown. All the simulations are performed for the range of actuator load moment *M* from 0 to 500 Nm.

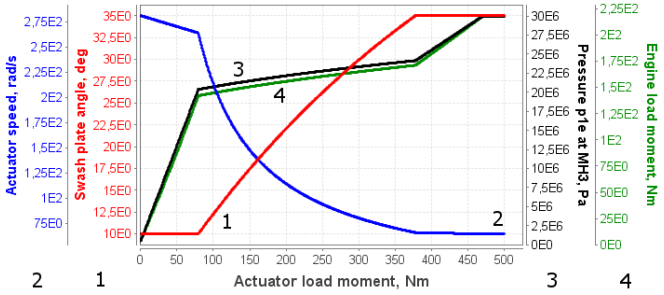


Figure 2: Graphs of hydraulic motor swash plate angle (1), actuator speed (2), pressure at hydraulic motor inlet (3) and engine load moment (4)

In Fig. 2 the position of hydraulic motor swash plate angle *al* (graph 1) is taken to change from 10 to 35 deg. Angle position depends on pressure drop in hydraulic motor through the automatic regulating system. The pressure drop in the hydraulic motor depends on the load moment to the actuator. In the range of the load moment *M* from 0 to 78.6 Nm the swash plate position angle remains at the initial value 10 deg. The angle increases from 10 to 35 deg, when the actuator load moment *M* changes from 78.6 to 377.5 Nm. The maximum pressure at hydraulic motor MH3 inlet (graph 3) *p*max = 30e6 Pa (set by safety valve) allows the transmission to operate until the load moment *M* = 469.5 Nm. The transmission is operable without automatic regulation when the load moment *M* changes from 377.5 to 469.5 Nm.

Actuator rotation speed (graph 2) drops from 281 to 263 rad/s (2683 to 2511 rev/min) linearly at lower load moments (*M* < 78.6 Nm). Further, the actuator speed exponentially drops in the area of automatic regulation. When the swash plate angle has taken *al* = 35 deg, the actuator rotation speed takes constant value ω = 65.4 rad/s (624 rev/min).

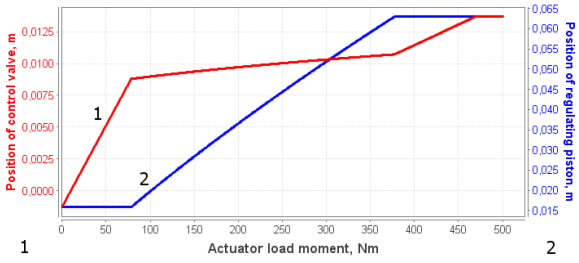


Figure 3: Graphs of control valve position (1) and position of the regulating piston (2)

Fig. 3 shows position of the control valve CV (graph 1) dependency on pressures at the left and right sides of the hydraulic motor.

Position of the regulating piston (graph 2) depends on the output pressure of the automatic regulating system (graph 2 of Fig.4) and determines regulating angle *al* of the swash plate. When the pressure of the automatic regulating system decreases, then the position of the regulating piston increases.

In Fig. 4 the diesel engine load moment (graph 4) increases and rotation speed (graph 3) drops by increasing the load moment to the actuator. Simultaneously the pressure at the pump outlet (graph 2) increases and the pump volumetric flow drops (graph 1).

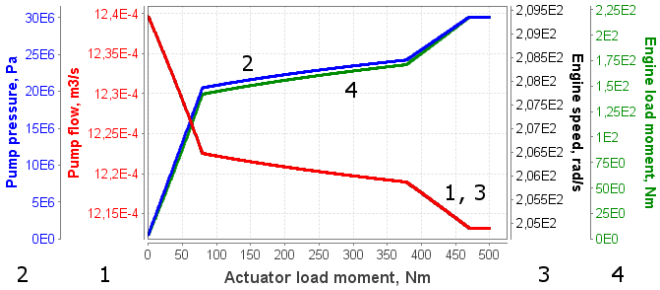


Figure 4: Graphs of engine load moment (4), engine rotation speed (3), pressure at the pump outlet (2) and pump volumetric flow (1)

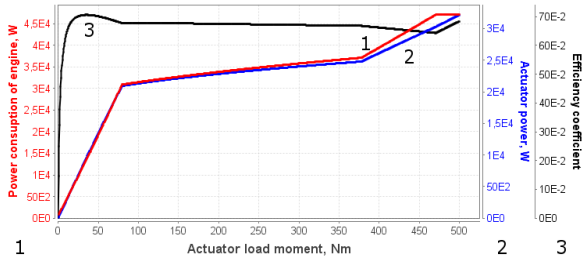


Figure 5 Graphs of power consumed by diesel engine (1), power of actuator (2) and efficiency coefficient of the transmission (graph 3)

Fig. 5 shows the power consumed by the diesel engine (graph 1) and the power delivered by the actuator (graph 2). Power consumed by the diesel engine (graph 1) reaches 31 kW at the beginning and 37 kW at the end of automatic regulation. Power of the actuator (graph 2) reaches 21 kW at the beginning and 25 kW at the end of automatic regulation. The power consumed by the diesel engine is 47 kW and power delivery of the actuator is 30 kW at actuator maximum load moment.

The efficiency coefficient of the transmission (graph 3) rapidly increases from 0 to maximum η = 0.70 at the load moment 35 Nm. In the range of automatic regulation (78.6 < M < 377.5 Nm) efficiency coefficient is almost constant η = 0.67. Later the efficiency coefficient drops to η = 0.64. At the actuator load moments higher than 469.5 Nm the transmission stops.

Observations

Simulated graphs of steady-state conditions are characterised by three breakpoints at three actuator load moment values.

- At actuator load moment 78.6 Nm at rotation speed 263 rad/s (2511 rev/min) the automatic regulation begins. The diesel engine output moment is 162.5 Nm at rotation speed 206.1 rad/s (1968 rev/min). The hydraulic pump delivers volumetric flow 12.2e-4 m^3/s (73.2 l/min).
- At actuator load moment 377.5 Nm at rotation speed 65.4 rad/s (624.5 rev/min) the automatic regulation ends. The diesel engine output moment is 196.4 Nm at rotation speed 205.4 rad/s (1961 rev/min). The hydraulic pump delivers volumetric flow 12.16e-4 m^3/s (72.9 l/min).

  To make the transmission operable at lower actuator rotation speeds volumetric flow of the hydraulic pump needs to be reduced. This can be achieved by decreasing the engine speed and/or by decreasing the hydraulic pump working volume.

- At the maximum pressure 30e6 Pa at hydraulic motor MH3 inlet the maximum actuator load moment is 469.5 Nm at rotation speed 64.3 rad/s (614 rev/min).The diesel engine delivers output moment 240 Nm at the rotation speed 204.5 rad/s (1953 rev/min) and the hydraulic pump delivers volumetric flow 12.1-4 m^3/s (72.6 1/min).

4.3 Simulation of Dynamics

Simulation task of dynamics of hydraulic open-loop rotary transmission with automatic regulation of a hydraulic motor is shown in Fig. 6.

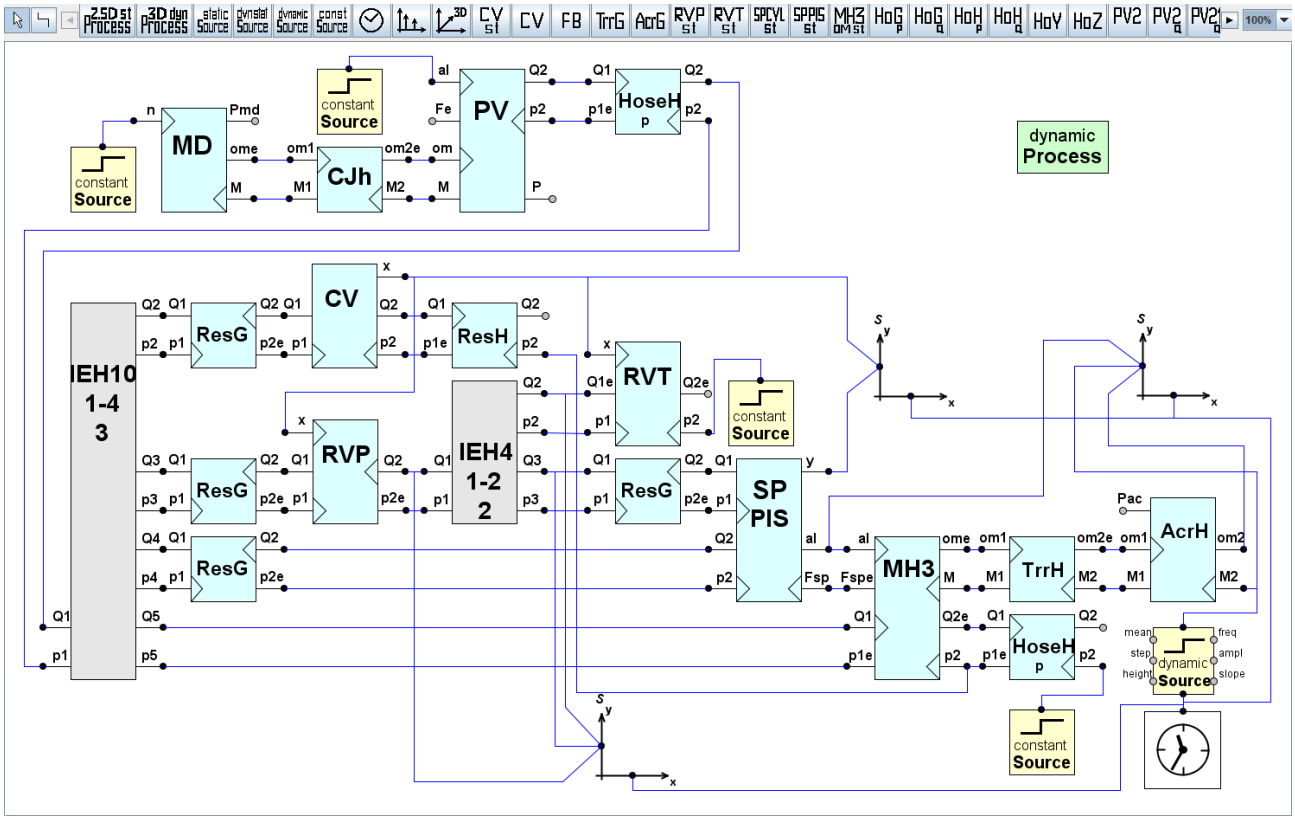


Figure 6: Dynamic simulation task of open-loop rotary transmission

In Fig. 6 the components of transmission are denoted: **MD** – diesel engine, **PV** – variable displacement hydraulic pump, **MH3** – variable displacement hydraulic motor, **HoseH_p** – hydraulic hoses which pressure p1e is iterated, **CV** – control valve, **RVP, RVT** – control valve inlet and outlet throttle edges, **SP_PIS** – hydraulic motor swash plate with regulating piston, **TrrH** – speed reducer, **AcrH** – rotation actuator, **IEH** – hydraulic interface elements, **ResG, ResH** – hydraulic resistors, **CJH** – clutch.

Additionally dynamic simulation task contains: **dynamic Process** – dynamic simulation manager, **dynamic Source** – dynamic input, **constant Source** – constant inputs, **clock**. The iterated variables are denoted with suffix "e".

The simulated graphs are shown in Fig. 7…10.

In Fig. 7 the step disturbance (graph 2) of the actuator input load moment (mean = 200 Nm, step = 20 Nm, rise time tmin =0.01 s) is shown. As a reaction to input disturbance, actuator rotation speed (graph 1) initially drops down, then begins to oscillate. The automatically regulated hydraulic motor swash plate position angle (graph 3) exponentially increases from 22.3 to 24.1 deg.

During transient responses oscillations of frequency ~6.4 Hz damping in ~1.5 s occur. All the oscillations in the graphs below are of the similar non-symmetrical character.

Frequency of the oscillations of the transmission mainly depends on the actuator inertial moment. The simulated dependence is shown in Fig.8. Damping of the oscillations is not influenced by the actuator inertial moment.

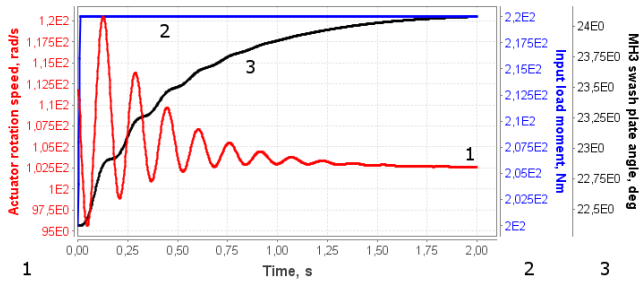


Figure 7: Graphs of input disturbance (2), actuator output rotation speed (1) and hydraulic motor swash plate angle (3)

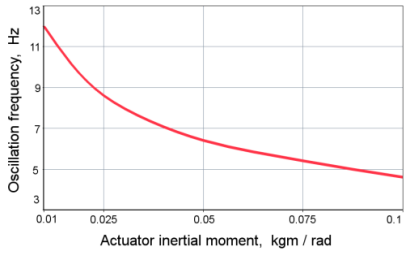


Figure 8: Dependence of the oscillation frequency on the actuator inertial moment

In Fig. 9 the dynamic processes of regulating system volumetric flows are shown. In the initial static position volumetric flows through RVP and RVT are equal and volumetric flow to SP_PIS is zero. Reacting to the step disturbance of the transmission load moment, servo-valve (CV, RVP, RVT) forces pressure to piston SP_PIS to change the

position angle of swash plate. Damped oscillations characterize the behaviour of volumetric flows. At the end of transient response volumetric flows return to their initial values.

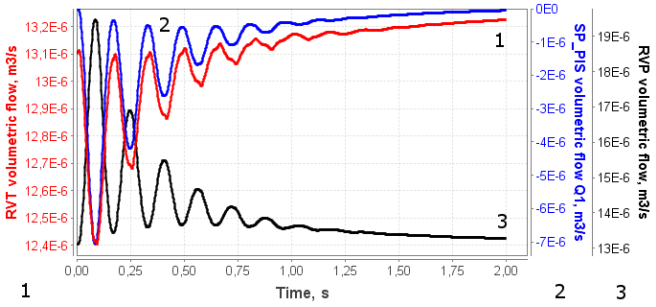


Figure 9: Graphs of volumetric flow of RVT (1), volumetric flow to hydraulic motor swash plate regulating piston SP_PIS (2) and volumetric flow of RVP (3)

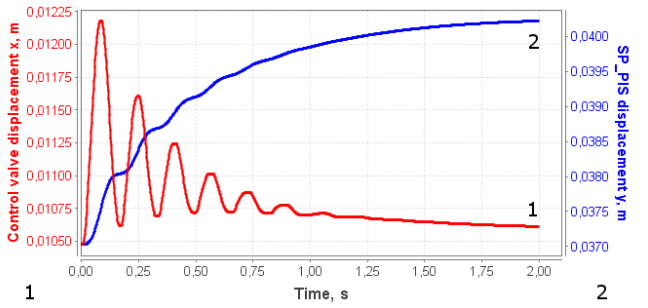


Figure 10: Graphs of displacements of control valve CV (1) and regulating piston SP_PIS (2)

In Fig. 10 the control valve CV takes a new position after damped oscillations (graph 1). The hydraulic motor swash plate regulating piston exponentially takes a new position (graph 2) from 0.0370 to 0.0402 m.

SUMMARY

In the paper simulation of steady-state conditions and dynamics of a hydraulic open-loop rotary transmission with automatic regulation of a hydraulic motor has been considered.

Mathematical multi-pole model of the transmission described in Part 1 of the paper are used as basis to describe simu-lation tasks and perform simulations.

Using a visual simulation environment CoCoViLa enables to describe multi-pole models graphically which facilitates the model development. Automatic synthesis of the calculation algorithms allows focusing on designing models of fluid power systems instead of constructing and solving simulation algorithms.

The simulation principles for steady-state conditions and for dynamics have been described. Main requirements were presented. The main parameters to be specified were shown. The graphical simulation tasks were presented and the results of simulations have been presented and discussed.

As a result of the simulation a configuration of an open-loop rotary transmission with automatic regulation of hydraulic motor has been proposed together with constructive and operating parameters. A great number of simulations have been performed to find a set of the best parameters.

Automatic regulating system proposed in the paper does not contain sensors and proportional valve. Therefore it is of a quite simple configuration and is expected to be more reliable and cheaper.

ACKNOWLEDGEMENTS

This research was supported by the Estonian Ministry of Research and Education institutional research grant no. IUT33-13, the Innovative Manufacturing Engineering Systems Competence Centre IMECC and Enterprise Estonia (EAS) and European Union Regional Development Fund (project EU48685).

REFERENCES

Grigorenko, P., A. Saabas and E. Tyugu. 2005. "COCOVILA – Compiler-Compiler for Visual Languages." *Proc. of the 5th Workshop on Language Descriptions, Tools and Applications, (Edinburgh, 2005),* v. 141, n. 4 of Electron. Notes in Theor. Comput. Sci., 137-142, Elsevier.

Grigorenko, P. and E. Tyugu. 2006. "Deep Semantics of Visual Languages." In: E. Tyugu, T. Yamaguchi (eds.) *Knowledge-Based Software Engineering. Frontiers in Artificial Intelligence and Applications*, vol. 140, IOS Press, 2006, 83 - 95.

Grossschmidt, G. and M. Harf. 2009. "COCO-SIM –Object-oriented multi-pole modelling and simulation environment for fluid power systems. Part 2: Modelling and simulation of hydraulic-mechanical load-sensing system*." International Journal of Fluid Power*,10(3),71 - 85.

Grossschmidt, G. and M. Harf. 2012. "Multi-pole modeling and intelligent simulation of technical chain systems, Part 2." *In: Proceedings of the 8th International Conference of DAAAM Baltic "Industrial Engineering",* Vol 2, 19-21st April 2012, Tallinn, Estonia; (Red.) Otto, Tauno, Tallinn; Tallinn University of Technology, 2012, 458-471.

Grossschmidt, G. and M. Harf. 2016. "Multi-pole modeling and simulation of an electro-hydraulic servo-system in an intelligent programming environment." *International Journal of Fluid Power,* Vol. 17, No. 1, March 2016, Taylor & Francis, 1 - 13.

Harf, M. and G. Grossschmidt. 2013. Multi-pole modelling and intelligent simulation environment for fluid power systems. *In: ESM '2013 : The 2013 European Simulation and Modelling Conference, Modelling and Simulation,* October 23-25, 2013, Lancaster University, Lancaster, UK, [Proceedings]: (Red.) Onggo, Stephanphan; Kavicka, Antonin. Ostend, Belgium: EUROSIS-ETI, 2013, 247 - 254**.**

Harf, M. and G. Grossschmidt. 2015. "Using Multi-Pole Modeling and Intelligent Simulation In Design of a Hydraulic Drive." *In: Proceedings of the 14h International Conference on Modeling and Applied Simulation,* September 21, 2015, Bergeggi, Italy, 6 p.

Kotkas, V.; A. Ojamaa; P. Grigorenko; R. Maigre; M. Harf; and E. Tyugu. 2011." CoCoViLa as a multifunctional simulation platvorm." *In: SIMUTOOLS 2011 - 4th International ICST Conference on Simulation Tools and Techniques,* March 21–25, Barcelona, Spain: Brussels, ICST, 2011, 1-8.

Murrenhoff, H. 2005. "Grundlagen der Fluidtechnik, Teil 1: Hydraulik, 4. neu überarbeitete Auflage." *Institut für fluidtechnische Antriebe und Steuerungen*, Aachen, 2005.

Parambath, J. 2016. "Industrial Hydraulic Systems – Theory and Practice." *Universal-Publishers Boca Raton,* Florida, USA, 2016.

PRODUCTION PLANNING

COUPLING OF RIGID BODY DYNAMICS WITH STRUCTURAL MECHANICS TO INCLUDE ELASTIC DEFORMATIONS IN A REAL-TIME CAPABLE HOLISTIC SIMULATION FOR DIGITAL TWINS

Max Krause[a], Kai-Uwe Schröder[a], Dorit Kaufmann[b], Tobias Osterloh[b], Jürgen Rossmann[b]

[a]Institute of Structural Mechanics and Lightweight Design, RWTH Aachen University
Wüllnerstrasse 7, 52062 Aachen, Germany
E-mail: {max.krause, kai-uwe.schroeder}@sla.rwth-aachen.de

[b]Institute for Man-Machine Interaction, RWTH Aachen University
Ahornstrasse 55, 52074 Aachen, Germany
E-mail: {kaufmann, osterloh, rossmann}@mmi.rwth-aachen.de

KEYWORDS
Rigid Body Dynamics, Structural Mechanics, Transfer Matrix Method, Overall System Simulation, Virtual Testbed, Digital Twin

ABSTRACT

The development of complex technical systems is a highly interdisciplinary process, requiring knowledge from various disciplines. To manage all facets of technical systems, holistic Virtual Testbeds are used as central simulation frameworks, providing cross-domain simulation capabilities. To realize these simulation capabilities, simulation algorithms have to be coupled adequately. Within our work, we present a new, highly efficient coupling of rigid body dynamics (RBD) and structural mechanics. The approach applies the transfer matrix method for the determination of the structural response of a system and is currently suited to beam structures. The real-time capabilities of this approach are pointed out and a robotics application is presented.

INTRODUCTION

Classical structural mechanics focuses on the analysis of single components, neglecting the fact, that these components are integrated into an overall system. To identify typical mechanical loads, structural mechanics are coupled to rigid body dynamics simulations, simulating the macroscopic systems behaviour. This results in a one-directional coupling from rigid body dynamics to structural mechanics, which is supported by various simulation tools. This one-directional coupling can be used for e.g. the performance analysis of various materials in an overall systems context (Kaufmann et al. 2017). Nevertheless, there are many applications where this one-directional coupling is not sufficient, as the structural deformations directly influence the dynamics of the overall system (e.g. aerospace engineering). To simulate these kinds of systems, a bidirectional coupling between rigid body dynamics and structural mechanics is required. Several theoretical approaches are known to realise this bidirectional coupling (Busch 2012, Stettinger et al. 2014). All these approaches require sophisticated mathematical methods and thereby claim substantial runtime.

By restricting the bidirectional coupling to beam structures, we derive a new, real-time capable coupling algorithm, which is based on the transfer-matrix-method. The taken restriction to beam structures is not a drawback, as many components in technical systems can be approximated as a beam. Thereby, this new approach allows analysing structural deformations in an overall systems context in a time-efficient manner.

HOLISTIC SIMULATION FOR DIGITAL TWINS

Based on this coupling, the range of validity of Digital Twins can be increased significantly. Digital Twins are virtual images of real world assets, describing all relevant physical properties of the real world asset (e.g. rigid body properties and structural mechanics properties). Additionally, Digital Twins include communication interfaces and control algorithms, e.g. a Digital Twin of a lightweight robot might consist of its kinematic model, its dynamics model, its structural mechanical model, its actuator models and its control algorithms. Nevertheless, a Digital Twin can never be all-embracing, there are always certain specific properties that will not be covered. It highly depends on the application the Digital Twin is used for.
Virtual Testbeds are used to analyse the interactive behaviour of a Digital Twin or a network of Digital Twins. Virtual Testbeds provide a software framework for the integration of arbitrary simulation algorithms and thereby can be seen as a run-time environment for Digital Twins (Schluse et al. 2017). For this reason, a Virtual Testbed is chosen for the implementation of the new, integrated approach of bidirectional coupling of rigid body dynamics and structural mechanics.

KEY METHODS

Rigid body dynamics algorithms are well suited to the simulation of the macroscopic dynamics of complex systems. Nevertheless, there are use cases, where local structural deformation needs to be considered. To calculate structural deformation analytical methods are used. To form the base for a comprehensive simulation approach for the coupling of rigid body dynamics and structural

mechanics, we first present the state of the art technologies from both worlds.

Rigid Body Dynamics

The dynamics of coupled rigid bodies are calculated according to the net external forces f_{ext} (e.g. actuated joints) and constraint forces f_c (e.g. bearing forces or contact forces). The inertia properties of the rigid bodies are described by the inertia matrix M.

$$M \cdot \ddot{x} = f_{ext} + f_c \qquad (1)$$

To determine the constraint forces f_c, the constraints are formulated in the velocity space. Holonomic constraints are formulated as $J \cdot \dot{x} = b$ and non-holonomic constraints are formulated as $J \cdot \dot{x} \geq b$, leading to a Linear Complementarity Problem (Baraff, 1996). J is the jacobian, describing the kinematics of the constraints, i.e. the relative velocity of two coupled bodies at a certain point, in certain direction vanishes. To numerically stabilize this approach, the Baumgarte stabilization is applied (Baumgarte, 1972).
For performance reasons an impulse based approach is applied to calculate the constraint forces (Jung 2011, Stewart and Trinkle 2000), where h is the step width of the simulation algorithm.

$$JM^{-1}J^T \cdot h \cdot \lambda + J[\dot{x} + M^{-1} \cdot h \cdot f_{ext}] = b \qquad (2)$$

Based on d'Alemberts principle, constraint forces are not allowed to introduce any energy to the system, the constraint forces can be directly obtained from the Lagrange multipliers λ.

$$f_z = J^T \cdot \lambda \qquad (3)$$

As the maximal coordinate approach directly calculates constraint forces, an efficient coupling of these forces to structural mechanics is easily accessible.

Transfer-Matrix-Method

The structural behavior of a structure idealized by e.g. a beam is always governed by the underlying differential equation. To calculate the displacements of an elastic structure based on the external forces, this differential equation has to be solved with the help of different kinds of methods. In contrast to the standard method, where the differential equation always needs to be solved based on the underlying boundary conditions, the transfer-matrix-method aims to derive a more general solution for the differential equation, which is valid for a high number of boundary and loading conditions. The derivation of such a solution is described for a beam structure in accordance to (Öry 1991):
A beam section with the bending stiffness EI of length L with its section forces Q_i and M_i and its external loading p is taken as the basis, see Figure 1.

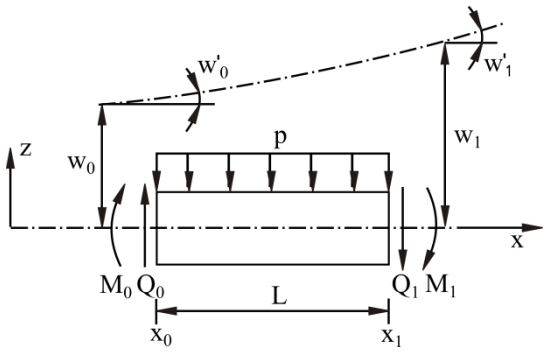


Figure 1: Displacements and cross section forces of a beam section (Öry 1991)

For a beam rigid in shear the differential equation and their fourth time indefinite integral in respect to x with the four constants C_i gets:

$$EI \cdot w(x)^{IV} = -p$$
$$EI \cdot w(x) = -\frac{p}{24} \cdot x^4 + \frac{C_1}{6} \cdot x^3 + \frac{C_2}{2} \cdot x^2 + C_3 \cdot x + C_4 \qquad (4)$$

It is assumed that all four state variables, the deformations w_0 and w_0' and the section forces M_0 and Q_0, at $x_0 = 0$ are known. They will be used in order to determine the four integration constants C_i. By using the knowns as boundary conditions in the left-hand side cross section, the integration constants follow:

$$C_4 = w_0 \cdot EI$$
$$C_3 = w_0' \cdot EI$$
$$C_2 = w_0'' \cdot EI = M_0$$
$$C_1 = w_0''' \cdot EI = Q_0$$

With the four integration constants, Equation (4) and the respective derivations, a relationship between the state variables at $x_0 = 0$ and $x_1 = L$ can be established:

$$w_1(x) = w_0 + w_0' \cdot x + \frac{M_0}{2 \cdot EI} \cdot x^2 + \frac{Q_0}{6 \cdot EI} \cdot x^3 - \frac{p}{24 \cdot EI} \cdot x^4$$
$$w_1'(x) = w_0' + \frac{M_0}{EI} \cdot x + \frac{Q_0}{2 \cdot EI} \cdot x^2 - \frac{p}{6 \cdot EI} \cdot x^3$$
$$M_1(x) = M_0 + Q_0 \cdot x - \frac{p}{2} \cdot x^2$$
$$Q_1(x) = Q_0 - p \cdot x$$

By inserting the identity, $1 = 1$, the equations can be represented in a matrix form, where the state vectors u_0 and u_1 are connected by a symmetric upper triangle matrix W, the so called transfer or field matrix:

$$\underline{u}_1 = \underbrace{\begin{bmatrix} 1 & x & \frac{x^2}{2EI} & \frac{x^3}{6EI} & -\frac{px^4}{24EI} \\ 0 & 1 & \frac{x}{EI} & \frac{x^2}{2EI} & -\frac{px^3}{6EI} \\ 0 & 0 & 1 & x & -\frac{px^2}{2} \\ 0 & 0 & 0 & 1 & -px \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\underline{\underline{W}}(x)} \underbrace{\begin{pmatrix} w_0 \\ w_0' \\ M_0 \\ Q_0 \\ 1 \end{pmatrix}}_{\underline{u}_0} \qquad (5)$$

The transfer matrix W, which is only a function of the bending stiffness EI, the length of the considered beam section and the outer loading, represents the solution of the differential equation of the beam. With this the state variables and therefore the displacements and section loads at the end, $x = L$, can be calculated in a very fast and stable manner by an ordinary matrix-vector-product. The disadvantage of this method is, that all state variables at one side have to be known or rather calculated before the transfer matrix method can be used. The successful application of the transfer matrix method is shown in many research activities, see for example (Rittweger 1992) or (Rittweger and Öry 1991). It was implemented into a stand-alone program call AstrA at the Institute of Structural Mechanics and Lightweight Design and used to design space launcher vehicles such as the Ariane V (Rittweger 2010).

Principle of virtual displacement

The principle of virtual displacement is the standard formulation of the Finite-Element-Method (FEM) and therefore a widely used method for solving mechanical problems. It is based on the principle of virtual work (Zienkiewicz and Taylor 2000). The approach will be demonstrated on a beam subjected to a load and a moment, see Figure 2.

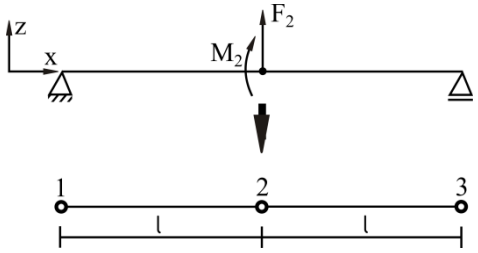


Figure 2: Statically determined beam subjected to a load

The beam is divided into sections/elements which lie between load introduction points, nodes 1, 2 and 3 in Figure 2. The principle of virtual work is used for one element. It states, that inner virtual work δW_i and external virtual work δW_e are equal (Nasdala 2010):

$$g = \delta W_i - \delta W_a = 0 \qquad (6)$$

For a rigid beam without line loading considering only bending the use of a polynomial 3rd order for the deformation w leads to the exact solution of the differential equation, see Equation (7).

$$w_j(x) = C_{0j} + C_{1j}x + C_{2j}x^2 + C_{3j}x^3 \qquad (7)$$

Using the polynomial for the different virtual systems j, which follow from introducing a virtual unit-displacement/unit-rotation for every degree of freedom, see Figure 3, the constants C_{ij} and therefore the deformation $w_j(x)$, also depicted in Figure 3, can be calculated.

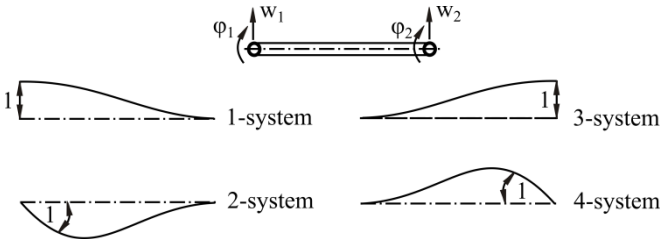


Figure 3: virtual systems of a beam element

Inserting these into the inner virtual work for every possible superposition of the virtual system i with system j the virtual work gets:

$$k_{ij} = \int_0^l EI \cdot w_i'' w_j'' dx \qquad (8)$$

Equation (8) represents the so called stiffness coefficients of the stiffness matrix for a beam element according to the principle of virtual displacement. Evaluating the principle of virtual work for two adjacent elements, see Figure 2, leads to a linear system of equations (Nasdala 2010):

$$\frac{2EI}{l^3}\begin{bmatrix} 6 & 3l & -6 & 3l & 0 & 0 \\ 3l & 2l^2 & -3l & l^2 & 0 & 0 \\ -6 & -3l & 12 & 0 & -6 & 3l \\ 3l & l^2 & 0 & 4l^2 & -3l & l^2 \\ 0 & 0 & -6 & -3l & 6 & -3l \\ 0 & 0 & 3l & l^2 & -3l & 2l^2 \end{bmatrix} \begin{Bmatrix} w_1 \\ \varphi_1 \\ w_2 \\ \varphi_2 \\ w_3 \\ \varphi_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ M_1 \\ F_2 \\ M_2 \\ F_3 \\ M_3 \end{Bmatrix}$$

As the stiffness matrix is singular, boundary conditions have to be inserted first before the system can be solved, meaning that the principle of virtual displacement is only valid for statically determined or overdetermined systems. Furthermore, with the method described displacements at specific points on the beam cannot be calculated without adding more elements. But this means, that the stiffness matrix gets particularly bigger and at some point equals the stiffness matrix of the Finite-Element-Method. Therefore losing its advantage, that is the calculation of the exact solution of the given problem.

CONCEPT OF THE COUPLING

The interface to the structural mechanics algorithms is completely integrated into the holistic simulation framework. This enables the user to include structural effects in every Digital Twin. To establish a real-time capability the coupling itself works on two levels: First, both structural calculation methods are combined in one algorithm. The algorithm is structured as slim as possible and finally implemented as a library, resulting in a very fast and efficient routine. Second, the library uses information obtained from the rigid body dynamics simulation as input variables to calculate the elastic displacements and stresses of the beam structure.

Coupling Transfer-Matrix-Method and Principle of virtual displacement

To develop a very fast and efficient calculation algorithm for the determination of the elastic deformations and the stresses of a beam structure, both above described

structural methods are combined. The calculation algorithm takes the external loads, boundary conditions and the bearing forces as input. Based on the external loads and the boundary conditions, the displacements and rotations at both ends of the beam structure are calculated using the principle of virtual displacement. With this information and the bearing or rather section forces all state variables at both ends of the beam structure are known. Further using the transfer matrix method, see Equation (5), the displacements, section forces and therefore stresses at requested points are calculated fast and efficiently. An example for the calculation of the state variables for multiple specific points is illustrated in Figure 4 and Equations (9) – (12).

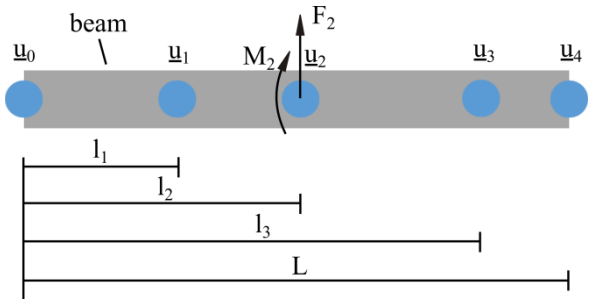


Figure 4: State vectors of a beam for specified points

$$\underline{u}_1 = \underline{\underline{W}}(x = l_1)\,\underline{u}_0 \quad (9)$$

$$\underline{u}_2 = \underline{\underline{W}}(x = l_2)\,\underline{u}_0 \quad (10)$$

$$\underline{u}_3 = \underline{\underline{W}}(x = l_3 - l_2)\,\underline{\underline{P}}\,\underline{\underline{W}}(x = l_2)\,\underline{u}_0 \quad (11)$$

$$\underline{u}_4 = \underline{\underline{W}}(x = L - l_2)\,\underline{\underline{P}}\,\underline{\underline{W}}(x = l_2)\,\underline{u}_0 \quad (12)$$

For the calculation beyond a load introduction point, e.g. the inertia forces at the centre of mass, a transfer matrix P for point loads needs to be considered, see Equation (11) and (12). This matrix amounts to:

$$\underline{\underline{P}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & M_2 \\ 0 & 0 & 0 & 1 & F_2 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The Coupling is illustrated in Figure 5.

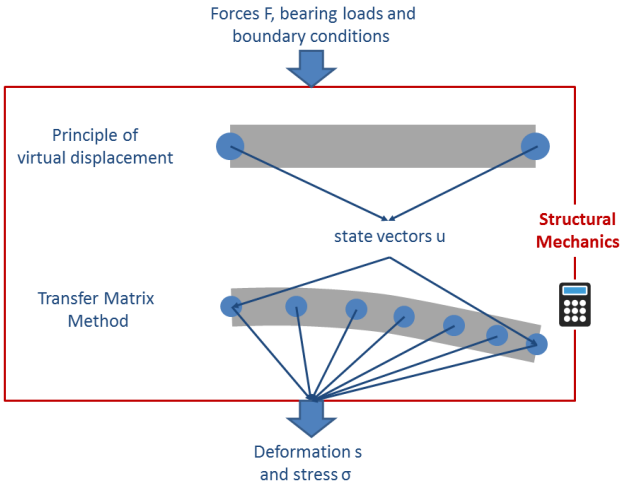


Figure 5: Concept of the coupling between both structural methods

With such a coupling between the structural methods the respective disadvantages of each method can be compensated. Furthermore, the calculation algorithm is significantly faster than the standard Finite-Element-Method (FEM) especially for a high amount of requested points/elements, because the most time consuming calculation steps consist of matrix-matrix products, matrix-vector products and "inverting" of the reduced stiffness matrix. In contrast to the FEM, where the stiffness matrix is directly proportional to the number of elements and has to be solved too, the stiffness matrix resulting from the principle of virtual displacement is a function of the number of basic sections only.

Coupling RBD and Structural Mechanics

One issue arises if the calculation algorithm is used as described above: The algorithm can only work in a real-time capable holistic simulation when the input variables are known beforehand. This arises from the fact, that the calculation of especially the bearing loads is a demanding task when the system is overly complicated, e.g. when a high number of external loads are present. The real-time capability of the algorithm would decrease. Therefore, a second coupling or rather the actual coupling between RBD and structural mechanics is performed in a sense as follow:

As many information as possible is calculated by the real-time capable rigid body dynamics and then used as input variables for the implemented structural mechanics calculation algorithm. In detail, the forces and torques f acting on the structure are measured via rigid body dynamics at currently three critical points: At both ends of the bar and at the centre of mass. The values and the possibly available boundary conditions are directly given to the structural algorithm. The algorithm treats the forces and torques at the centre of mass as external loads. The forces and torques at the ends of the beam are treated either as external forces, if no boundary condition is present at the end, or as bearing loads for the case of boundary conditions present. Following, the algorithm calculates the deformations (translation and rotation in six dimensions) and stresses for each element along the bar and outputs them to the rigid body dynamics simulation (RBD). The resulting deformations are visualized and an intuitive colour coding is overlaid. Figure 6 shows the exchanged variables and the coupling of the two simulation methods.

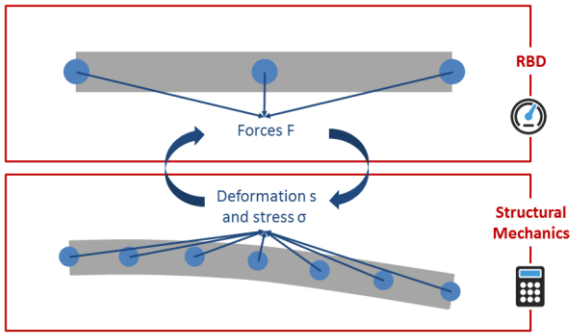


Figure 6: Concept of the coupling between RBD and Structural Mechanics.

APPLICATION

As a first application scenario for the developed coupling an example from robotics was chosen. A lightweight robot performs a simple task. During the movement, it comes into contact with one end of a bar, which is fixated on the other side. The robot itself is described by its rigid body model, whereas the bar is modelled using the coupling methodology described herein. This example shows how the integration of structural mechanics smoothly fits into the holistic simulation (see Figure 7).

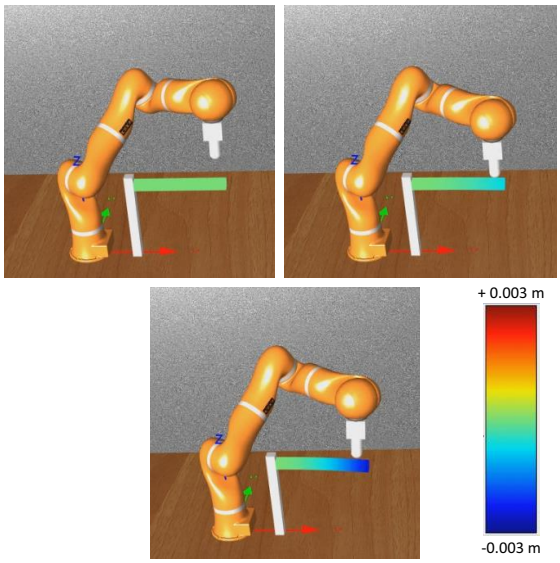


Figure 7: Example of the developed coupling: A lightweight robot causes deformation in a fixated bar when it is pressing onto it (colour coding: displacement in z-direction).

CONCLUSION AND OUTLOOK

In this paper, a new and real time capable coupling method between rigid body dynamics and structural mechanics for the inclusion of elastic deformations in a holistic simulation was presented. The method developed and presented led to promising results, especially in regards to the real-time capability. It was shown that the transfer-matrix-method with its basic matrix-vector operations smoothly fits into the coupling mechanism. Nevertheless, several restrictions with regard to the applicability have to be accepted. The most important being the necessity of a statically determined system, which is a prerequisite for the principle of virtual displacement. It must be noted that this restriction does not apply to the transfer-matrix-method. To allow a more general application future work should first focus on two aspects: First being the extension of the coupling methodology to allow the simulation of statically undetermined systems. This could be achieved by calculating the rigid body motion of the system with the help of the RBD and then superimpose it with the deformation obtained for an appropriate statically determined system using the method of the omega function, see (Horst 1992). Second, the deformation obtained from the calculation algorithm should be used to update the inertia matrix in the rigid body simulation to account for changes in the loading condition due to the deformation of the beam. Hereby, a comprehensive study on the possible violation of the principle of conservation of energy and a derivation of an appropriate solution should be conducted.

REFERENCES

Baraff, D. 1996. “Linear-Time Dynamics using Lagrange Multipliers.” *In Proceedings of SIGGRAPH 96,* 1996, (New Orleans, Aug 4-9).

Baumgarte, J. 1972. “Stabilization of constraints and integrals of motion in dynamical systems” *Computer Methods in Applied Mechanics*, p. 1-26, 1972.

Busch, M. 2012. “Zur Effizienten Kopplung von Simulationsprogrammen“, *Ph.D. Thesis*, *University Kassel*, 2012.

Horst H. 1992. *Vorlesungen über lineare Stabstatik*. Institut für Statik. Seeliger Sofort-Druck. Hannover.

Jung, T. 2011. “Methoden der Mehrkörperdynamiksimulation als Grundlage realitätsnaher Virtueller Welten.” *Ph.D. Thesis, RWTH Aachen,* 2011.

Kaufmann, D., Rast, M., Rossmann, J. 2017. „Bridging the Gap between Overall System Simulation and Finite Element Analysis – An eRobotics Approach”, *In Proceedings of ASTRA 2017*, (Leiden, Netherlands).

Nasdala L. 2010. *FEM-Formelsammlung Statik und Dynamik*. Vieweg+Teubner, Wiesbaden.

Öry, H. 1991. “Structural design of aerospace vehicles I”. Institut für Leichtbau RWTH Aachen University. Aachen.

Rittweger, A.. 1992. “Statik, Stabilität und Eigenschwingungen anisotroper Rotationsschalen beliebigen Meridians mit der Übertragungsmatrizen-Methode”. *Ph.D. Thesis* RWTH Aachen University.

Rittweger, A.. and Öry, H. 1991. “Stability-analysis of elastic shells of revolution with the transfer-matrix-method – a fast and reliable approach based on the exact solution of the shell equation”. Buckling of shell structures, on land, in the sea, and in the air. Spon Press.

Rittweger, A.; Christianson, S. and Öry, H. 2010. “Dimensioning of orthotropically stiffened CFRP shells of large launch vehicles for load introduction and stability.” In *International journal of structural stability and dynamics 10*, No. 4, 601-621.

Schluse, M., Atorf, L., Rossmann, J. 2017. “Experimentable Digital Twins for Model-Based Systems Engineering and Simulation-Based Development.” *In Proceedings of SysCon 2017*, 2017 (Montreal, Canada, April 24-27).

Stettinger, G., Horn, M. Benedikt, M., Zehetner, J. 2014. „Model-based Coupling Approach for non-iterative Real-Time Co-Simulation“, *In Proceedings of European Control Conference (ECC)*, 2014, France.

Stewart, D. and Trinkle J.C.. 2000. “An Implicit Time-Stepping Scheme for Rigid Body Dynamics with Coulomb Friction.” In IEEE International Conference on Robots and Automation, 2000, p 162 – 169.

Zienkiewicz O.C. and Taylor R.L. 2000. *The-Finite-Element-Method fifth edition Volume1: The Basis*. Butterworth-Heinemann. Jordan Hill. Oxford.

Process-based Ship Production Planning System Modeling for Supporting Decision Processes

Dongsu Jeong
Department of Industrial & Management Engineering
Korea University
Seoul, Republic of Korea
E-mail: jdsvs2979@korea.ac.kr

Yonnho Seo
Department of Industrial & Management Engineering
Korea University
Seoul, Republic of Korea
E-mail: yoonhoseo@korea.ac.kr

KEYWORDS
Shipbuilding, Ship production system, Ship production planning, Process-based Modeling & Simulation, Cyber-Physical System

ABSTRACT

For a long time, shipbuilding industry has experienced suffering in the ship production from unexpected problems such as short-term delivery of ships, schedule changes, and deteriorating weather conditions, etc. Even though many researchers carried out various studies to improve their competitiveness as well as to detect problems in advance and present immediate solutions, it is still unresolved and very critical to maintain a cutting edge productivity of shipbuilding companies. In this paper, we propose a process-based production planning method to be capable to generate and evaluate a best alternative to an unexpected problem in ship production. The best alternative is generated, simulated and analyzed through a consecutive 4-steps, i.e., constraint structure modeling, work dispatching modeling, process-based modeling & simulation, and simulation-based analysis.

1. Introduction

Recently, the shipbuilding industry has had many difficulties in building ships due to the sharp decline in ship contracts caused by the global financial crisis. Many researchers are studying to be able to increase the competitiveness of ship production by using new technologies such as ICT (Information and Communication Technology) and CPS (Cyber-Physical System). And, research on production system that can reduce production costs through efficiencies of ship production is becoming important (Baek et al. 2017). Generally, shipbuilding is single (fixed) product production & bay production, also traditional labor-intensive assembly industry. In shipbuilding, there are many types of ship production planning such as block division, assembly process, outfitting process, block erection process, and etc. Among these production process activities, the assembly process is very important as it determines what will happen in other process activities and also spends many time and cost in ship production. Therefore, many researchers are studying production systems that can effectively manage the assembly process by reducing the time and cost the shipboard, assembly schedule, and logistics schedule and etc. (Cho et al. 1996; Storch et al. 1988).

In reality, however, the shipbuilding environment has an unexpected abnormal situation such as ship's short delivery, change of production schedule, defective products and deteriorating weather conditions and etc. Currently, researchers of shipbuilding are trying to solve abnormal situations and problems by changing the production schedule of ship, changing the production process, adding more workers, and etc. However, it is difficult to consider all production schedules between other production processes. Furthermore, research suggesting immediate solutions to abnormal situations and unexpected problems is very minor in the shipyard, and shipyards have many constraints due to the characteristics of the shipbuilding industry and differentiated production processes. Despite the development of ICT and the implementation of Big Data & AI (Artificial Intelligent) technologies, it need to takes much research and time to apply them to the actual shipbuilding industry (Jeong et al. 2016). Therefore, shipbuilding needs theory and solution that can support the best decision making by detecting problems with various abnormal situations, and it needs CPS-based autonomic system which can integrate actual process situation and virtual simulation information through IoT and simulation.

In this paper, we proposes a process-based production planning method that can generate and evaluate a best alternative for supporting decision making such as a new schedule and process plan by pre-testing and predicting problems arising from abnormal situations at large assembly processes.

2. Manufacturing Processes and Production Planning of Shipbuilding

2.1 Manufacturing Processes of Shipbuilding

The shipbuilding industry carries out tasks that consider load balancing of the workplace, loads, and facilities in method of single (fixed) product production & bay production until the ship is ordered and delivered. Manufacturing processes of shipbuilding such as Figure 1 are produced through sequential processes such as unloading & storage of steel, steel cutting, sub-assembly, middle-assembly and large-assembly according to works required ship construction, and then ship is carried out by repeating movement and assembly of blocks until erection on the dock after outfitting, painting in shop, PE(Pre-Erection).

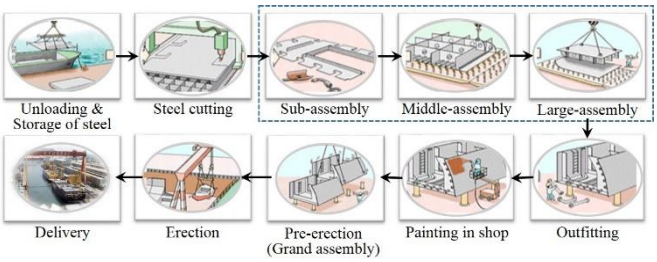


Figure 1: Procedure of ship manufacturing process (adapted from Korea offshore & shipbuilding Association)

First, the shipbuilding process begins with a block division. Generally, ships are divided into hundreds of blocks of moderate size. A block consists of several unit blocks, and each unit block is composed of steel plates and sections that are cut or bent by cutting and bending. In particular, the hull block assembly process is the most important for planning the block assembly process since it is about 48% to 50% of the shipbuilding process (Cho et al. 1996). The block assembly consists of the sub-assembly, middle-assembly and large assembly at in-house shop, and the painting and outfitting such as pipes and electrical work are subsequently performed, and then the grand assembly is created through the pre-erection. Finally, grand assembly is built as a ship using a very large crane (Zip crane or Goliath crane) in the main-dock. Thus, the ship is carried out in accordance with the shipbuilding manufacturing process, and the block assembly is performed using various equipment (welder, transporter, crane and etc.).

Nowadays, shipbuilding manufacturing process is difficult to consider various constraints (block relationships and characteristics, block type, timekeeping, weather and etc.) and factors (human, equipment utilization and etc.) used in ship production, and there is also a lock of quantify research (Seo and Park 2003). In addition, in order to improve ship quality and decrease ship demand due to changes in the shipbuilding world market, the shipbuilding manufacturing production system that can systematically analyze ship manufacturing processes and identify various problems in advance is very important. Therefore, new system which can identify the flexible relationship between the product-process-resource required for the shipbuilding manufacturing process and then manage the entire shipbuilding production process as layered detail is needed (Lee 2013).

2.2 Production Planning of Shipbuilding

The manufacturing processes of shipbuilding are carried out based on the production planning. The manufacturing processes of shipbuilding operate production plan (ship table planning, long-term planning, mid-term planning, short-term planning and execution planning) associated with 5 stages (Company, Plant, Factory, Cell and Resource Level). Production planning, the central task of connecting design and production of ship, determines the blocks are assembled and erected, the work schedule, and the spatial batch plan, as well as production planning is to be the basis for the use of equipment, human resources placement operation, material supply and demand planning. Production planning proceeds with the design and production of ship in accordance with the 'facility-fixed strategy' that does not change a given production facility or equipment, and production planning should develop facilities utilization plans to meet the product's delivery dates and production capabilities. Also, production planning should provide critical information to support decision making about management, such as strategies and vision of company shown in Figure 2, as well as to increase the efficiency and productivity of production systems. Therefore, the production planning must predict and present a new production system suitable for the changed shipbuilding environment (Lee 2013 and Woo 2005).

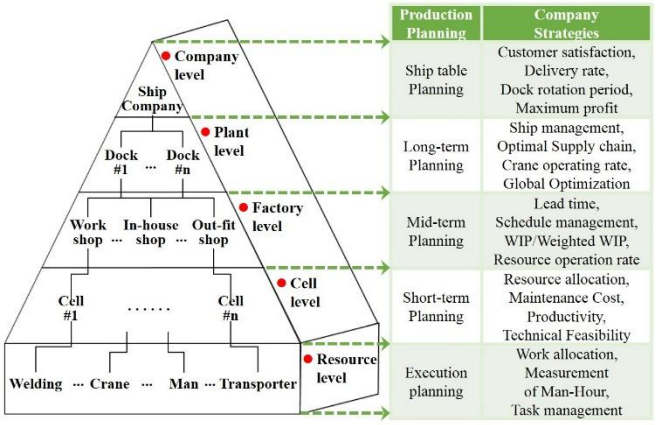


Figure 2: Hierarchy of ship production planning and company strategies

The general ship production planning begins with the overall plan for ship table planning. Afterwards, the ship table plan is subdivided to show the long-term planning, mid-term planning, and short-term planning considering of capacity plan. Each planning has key information that should be considered according to a specific period of ship production, and includes various information for ship production at the shipyard. Table 1 shows explanation of each planning, and most shipyards have similar phase of production planning such as Table 1. However, these production planning procedure raises many complex problems due to various constraints and complex construction of various shipbuilding divisions, and the complex production plans are causing many disruptions to shipbuilding production (Song et al. 2009). To address these problems, research was done to support decision making or to plan production in advance, but no clear solution was given (Sladoljev 1996).

Table 1: Ship production planning (adapted from Song. et al., 2009)

Phase	Contents
Ship table planning	To determine the ship product mix, To determine batch duration (dock rotation period), To determine the main key-events of the each planned ship
Long-term planning	Load distribution along the ship product mix of the master planning, To determine the capacity of the available human resources, To determine the capacity of the facility type resources
Mid-term	To check or validate the master and long-

planning	term planning, To schedule mid-term activities
Short-term planning	To check or validate the mid-term planning, To schedule short-term activities
Execution planning	To analyze worker allocation, work flow, work order, and output of the execution operation

Many shipyards are using M & S (Modeling & Simulation), one of the most efficient methods of predicting the future, to carry out production planning and processes. M & S is a scientific prediction technology that can improve the accuracy of the production plan. In order to integrate software and systems, it is necessary to support systematic development methodology, programming techniques and tools (Andritsos 2000). Many shipyards are currently implementing various systematic approaches and projects to effectively apply simulation technology into production planning.

2.3 Previous research review

Currently, most large shipyards use production scheduling systems to establish production plans that take into account process and plant loads. However, the functions supported by the production planning system can't take into consideration various physical constraints of the production site, so that the production planning is delayed, causing delays in delivery and waste of resources.

In order to solve these problems, shipyards are using simulation to verify ship production plan considering various constraints of shipbuilding process and work breakdown structure of block (Woo and Song 2014). Woo (2005) proposed a simulation modeling method of module concept for shipbuilding process during shipbuilding process (Woo 2005) and studied of production system using heuristic algorithm and CBR (Case-based Reasoning) for optimal block assembly sequence and production plan (Sheen and Seo 2017; Seo, et al., 2007; and Seo and Park 2003). However, existing commercial software has limitations in expressing complex shipyard facilities and plans. Han et al. (2008) modeled and simulated shipboard panel lines using Petri net, which is a process-based modeling method considering the characteristics of the shipbuilding process (Han et al. 2008). There was a limit to expanding to the whole process of the shipyard. After that, Lee et al. (2014) analyzed the production information of the shipyard and studied PCM (Process-Centric Modeling), which is a method of constructing shipyard simulation model based on process (Lee et al. 2014). Lee et al. (2014) insisted that the process-based modeling methodology considering the production environment of the shipyard is appropriate because the production management system of the shipyard is managed based on the process. However, previous studies have focused on the process of constructing the model rather than performing the simulation. But there are limitations to the research that supports the immediate solution and decision making using simulation (Lee et al. 2014).

In this paper, we propose a simulation system design based on ship manufacturing process including BLK design information and resource information as information to support decision making in case of a problem in case of abnormal situation along with the methods proposed in previous studies.

3. Model of Process-based Ship Production Planning System

Figure 3 illustrates the schematic diagram of a production planning system that supports real-time decision making, which may be needed on unexpected problems in ship production planning. This paper presents a process-based ship production system which is modeled in multi-stage to solve the abnormal situations and problems that may occur at each level. A process-based ship production system for ship production consists of 4 modules: "Constraint Structure Model," "Work Dispatching Model," "Process-based Modeling & Simulation," and "Simulation-based Analysis." Then, these sub systems cooperates to support the decision making of shipbuilding production such as process planning, resource utilization, block re-scheduling and etc.

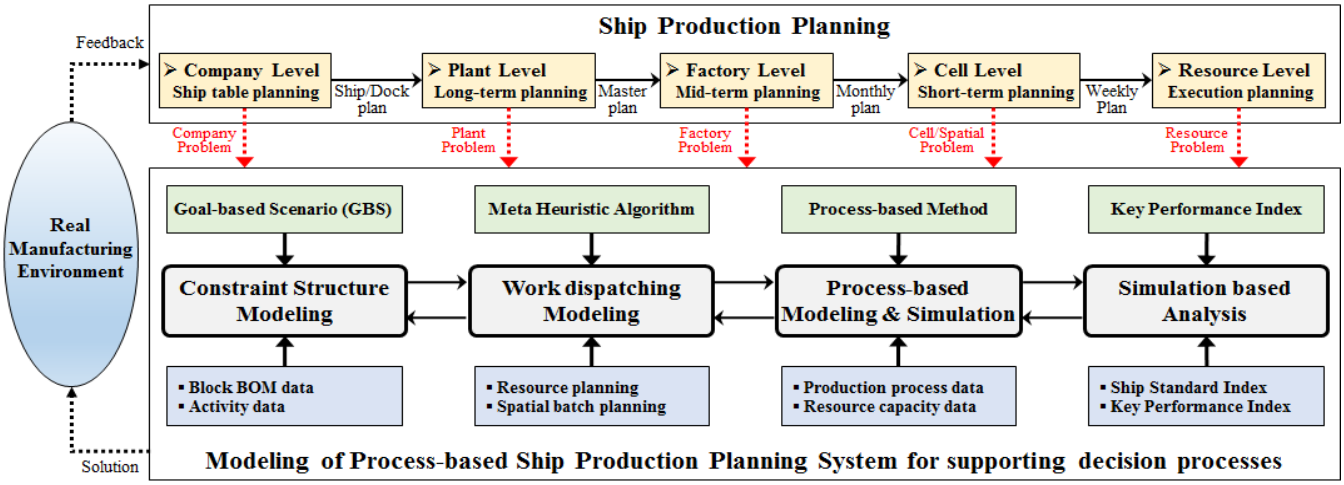


Figure 3: Concept of Process-based Ship Production Planning System Modeling

3.1 Constraint Structure Modeling

A constraint structure is modeled to generate alternatives that can solve problems by collecting and analyzing abnormal information in a ship hull production process. First, as shown in Figure 4, a process plan for a block can be generated

through BOM (bill of material) information and Activity information of the block required for ship production. The process plan for a block shows the production schedule and constraint information corresponding to the block by combining the production information of the sub-block and the parts of the block with the activity schedule data in monthly planning level. The process plan information can be utilized to confirm the resource level required for the block production and to comprehensively manage the whole production planning system.

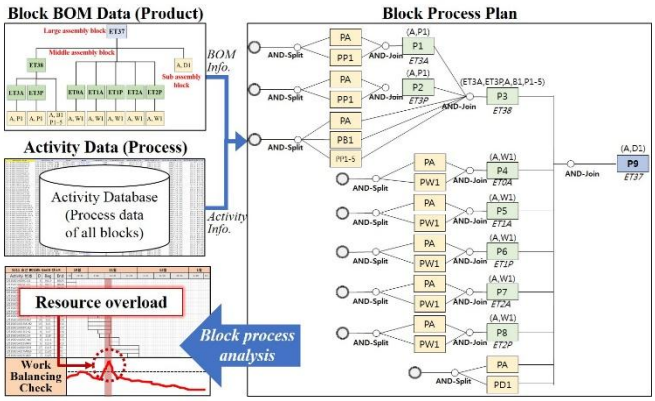


Figure 4: Generation of block process plan through bock BOM and Activity Data

Then, based on the block process plan data, the constraint structure according to a GBS (Goal Based Scenario) methodology is designed. As shown in Figure 5, the GBS is a methodology for generating multiple solution alternatives by combining components to achieve a goal given. Firstly, the following 6 elements are considered to create a solution alterative to resolve an abnormal situation during ship hull production processes: goal, workplace, block, resource, schedule, and objective function. Various scenarios are created by combining components one for each of 6 elements. Then, these scenarios are evaluated through the "Work Dispatching Modeling" and "Process-based Modeling & Simulation" procedures, and the best alterative among alternatives in scenario is selected.

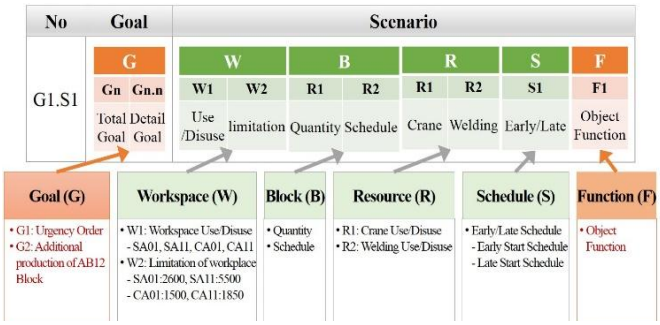


Figure 5: Design of Constraint Structure using GBS

3.2 Work Dispatching Modeling

Work Dispatching module is designed to develop the schedules of blocks that are used in the assembly shop based on the solution alternatives and corresponding information provided by a Constraint Structure Modeling module. This module exploits the possibility to reduce the completion time of assembly work in a spatial resource, by using the more resource (or cost) to the process of block assembly. That is, some solution alternatives are selected in this module by scheduling block assembly processes with satisfaction of the constraints from the Constraint Structure modeling. As shown in Figure 6, Work Discharging Modeling designed mathematical model by combining the large assembly block schedule and the spatial batch schedule, the program was executed using GA (Genetic Algorithm) in order to quickly derive the appropriate schedules within an allowable time.

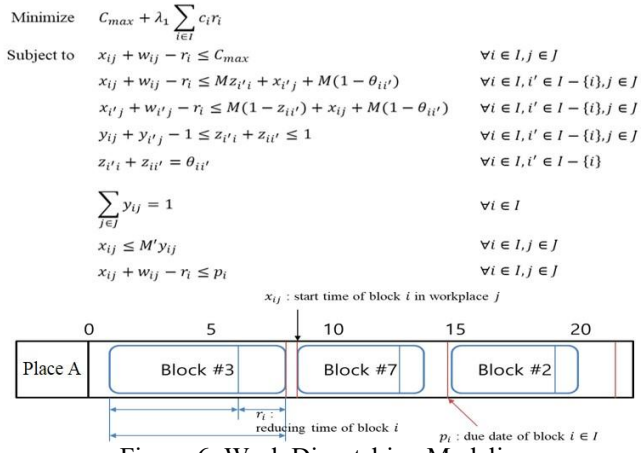


Figure 6: Work Dispatching Modeling

3.3 Process-based Modeling and Simulation

Process-based modeling and simulation is a module to evaluate and select alternatives based on scenarios, to meet the constraints on resource utilization and production feasibility of the block assembly works. As shown in Figure 7, the terms "Unit model", "Integrated Process Network", and "Process-based simulation" are introduced to respectively express a process plan for a block, a process network to integrate the processes for blocks in an assembly plant and the executable process model converted from integrated process network. First, a unit model means a process plan model for a block by integrating information on product, process, and resource about the block assembly processes. Integrated process network to express the total processes for an assembly plant is designed by integrating unit models and processes that each unit model has with its schedule, according to the schedule of the work dispatching algorithm. A IPN (Integrated Process Network) algorithm that link individual processes into one integrated process network is developed. The integrated process network resulted from the PNI is converted into a process model which can be executed by simulation. The process-based modeling method is a modeling and simulation method that can express the specific performance and functions performed on each node in a process by redefining the forward and backward relationships of systems and processes (Jeong and Seo 2015). Then, in this research, process-based simulation for a solution alternative is conducted to evaluate the alternative by using a Petri-Net simulation tool.

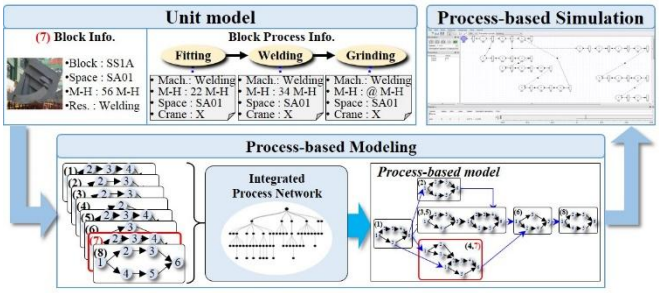


Figure 7: Process-based Modeling & Simulation

3.4 Simulation based Analysis

Simulation-based analysis provides a base to evaluate alternatives in terms of KPIs (key performance indexes) to maximize the company profits. As shown in Figure 8, the comparative analysis for the alternatives can be possible by using the evaluating results from process-based simulation. The simulation-based analysis evaluates the alternatives for each scenario. The assessment is performed through the standards of KPIs and production analysis for the shipbuilding. It provides the most appropriate alternatives for abnormal situations, based on the KPIs of the company and the current states of the processes.

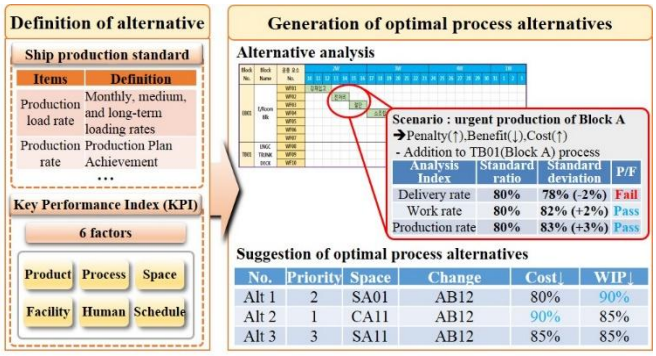


Figure 8: Simulation based Analysis

4. Conclusion

Research for supporting decision making according to abnormal situations in shipbuilding environment is very important. The shipbuilding production requires advanced system that can comprehensively analyze diverse information because of complex problems many different constraints and processes. In this paper, we propose a design of process-based ship production process system modeling including information such as Block, Facility, and Schedule to support decision making from abnormal situation and problems at large assembly processes, which are the major processes of the ship production. As a result, the system presented in this paper can provide information to support decision making based on the relations among sub-systems. Further research is being carried out to implement the system.

5. Acknowledgement

This work was supported by National IT Industry Promotion Agency (NIPA) grant funded by the Korea government (MSIP) (S0607-18-1004, the development of the Shipyard IHOP process standard simulator and decision support system)

REFERENCES

Andritsos, F. and Perez-Prat, J. 2000. *The automation and integration of production processes in shipbuilding*. State-of-the-Art report, Joint Research Centre. European Commission, Europe.

Baek, S. J.; Oh, D. K.; Lee, D. K.; Lee, P. L.; Ryu, C. H.; Woo, J. H.; and Jeong, Y. K. 2017. *A Process-centric Simulation Modeling Method Improving Product, Process, and Facility Information Representation Method*. Journal of the Korean Society of Marine Environment & Safety, 23(4), 421-431.

Cho, K. K.; Lee, S. H.; and Chung, D. S. 1996. *An automatic process-planning system for block assembly in shipbuilding*. CIRP Annals-Manufacturing Technology, 45(1), 41-44.

Han, S. D.; Ryu, C. H.; Shin, J. G.; and Lee, J. K. 2008, *Modeling and Simulation of Ship Panel-block Assembly Line Using Petri Nets*. Korean Journal of Computational Design and Engineering, Vol. 13, No. 1, pp. 36-44.

Jeong, D. and Seo, Y. 2015. *Process-based Modeling for Performance Evaluation of Weapon Systems*. Asia Simulation Conference 2015

Jeong, Y. K.; Woo, J. H.; Oh, D. K.; and Shin, J.G. 2016. *A Shipyard Simulation System using the Process-centric Simulation Modeling Methodology : Case Study of the Simulation Model for the Shipyard Master Plan Validation*. Transactions of the Society of CAD/CAM Engineers, 21.2(2016): 204-214.

Lee, D. G. 2013. *Study on the PPR3-S information-based neutral model and system for integration and extension of shipbuilding production planning simulations*. Doctoral dissertation, Ph. D. Thesis. Seoul National University.

Lee, D. K.; Kim, Y.; Hwang, I. H.; Oh, D. K.; and Shin, J. G. 2014, *Study on a process-centric modeling methodology for virtual manufacturing of ships and offshore structures in shipyards*. The International Journal of Advanced Manufacturing Technology, Vol. 71, pp. 621-633.

Seo, Y. and Park, C. 2003. *Machine selection considering workload balance at the parts fabrication stage of ship hull construction*. International journal of manufacturing technology and management, 5(1-2), 81-90.

Seo, Y.; Sheen, D. and Kim, T. 2007. *Block assembly planning in shipbuilding using case-based reasoning*. Expert Systems with Applications, 32(1), 245-253.

Sheen, D. and Seo, Y. 2017. *Nesting and Berth Allocation by an Expert System Using Heuristic Search*. Journal of Software Engineering and Applications, 10(04), 311.

Sladoljev, Z. 1996. *Search for a model of effective ship production management*. Journal of ship production. 12(4), 220-229.

Song, Y. J.; Woo, J. H.; and Shin, J. G. 2009. *Research on a simulation-based ship production support system for middle-sized shipbuilding companies*. International Journal of Naval Architecture and Ocean Engineering, 1(2), 70-77.

Storch, R. L.; Hammon, C. P.; and Bunch, H.M. 1988. Ship production.

Woo, J. H. 2005, *Modeling and Simulation of Indoor Shop System of Shipbuilding by Integration of the Product, Process, Resource and Schedule Information*. Doctoral dissertation, Ph. D. Thesis. Seoul National University.

Woo, J. H. and Song, Y. J. 2014. *Systematisation of ship production management and case study for ship block assembly factory*. International Journal of Computer Integrated Manufacturing, 27(4), 333-347.

BIOGRAPHIES

Dongsu Jeong is currently a Ph.D. candidate at the Department of Industrial & Management Engineering from the Korea University, Korea. He has researched on development of ship production planning system for shipbuilding Industry 4.0s among the projects of the National IT Industry Promotion Agency (NIPA) of Korea. His current research interests are the process modeling & simulation and intelligent manufacturing system.

Yoonho Seo is currently a professor of Industrial & Engineering at the Korea University, Korea. He holds an M.S. and Ph.D. in Industrial Engineering from the Pennsylvania State University. Dr. Seo has recently researched on development of ship production planning system for shipbuilding Industry 4.0s among the projects of the National IT Industry Promotion Agency (NIPA) of Korea. His primary research interest involves intelligent manufacturing system design, intelligent modeling & simulation, computer-aided process planning, etc. He has published approximately 40 papers on these and related subjects.

DEVELOPMENT, IMPLEMENTATION AND EVALUATION OF A COMPLEXITY MEASURE FOR THE WORK OF ASSEMBLY TEAMS IN ONE-PIECE-FLOW ASSEMBLY SYSTEMS EMPLOYING SIMULATION STUDIES

Benedikt A. Latos, Peyman Kalantar, Florens L. Burgert, Matthias G. Arend,
Verena Nitsch, Philipp M. Przybysz Susanne Mütze-Niewöhner
Institute of Industrial Engineering and Ergonomics
RWTH Aachen University
Bergdriesch 27, 52062 Aachen, Germany
E-mail: b.latos@iaw.rwth-aachen.de

KEYWORDS
Production, Assembly planning, Complexity, Discrete simulation, General Systems Theory

ABSTRACT

The growing demand for individualized consumer goods and the increase in production volume variations raise the level of complexity in assembly systems. As a result, people that work in such assembly systems have to cope with the flexibility requirements with respect to the assembly processes. The first step in complexity management is to measure the level of complexity. This paper presents the development of a complexity measure for the work of assembly teams in One-Piece-Flow assembly systems. The measure is conceptually developed and implemented into a 3D simulation model[1] of a One-Piece-Flow system. Furthermore, simulation studies are conducted to evaluate the plausibility of the complexity measure. The results indicate that the measure is able to capture the complexity of the work of assembly teams. Furthermore, initial results of a practical application of the measure are presented.

INTRODUCTION

The digital transformation forces companies to rethink their products and value creation processes. In the context of advanced manufacturing, innovative technologies are used to enable the production of individualized products that are increasingly equipped with mechatronic components (Schuh et al. 2017). This causes an increase not only in product complexity, but also in the structural complexity of companies (e.g. the number of processes).
The companies are confronted with different challenges during these developments: permanent product innovation pressure, rapidly changing and hardly predictable customer requirements, a continuously rising number of variants as well as strongly fluctuating sales figures (cf. Spath et al. 2013). Hence, producing companies are subject to highly competitive environments in an increasingly dynamic and volatile market. This calls for a short cyclical and rapid adaption of production processes and structures. Current efforts in the automotive industry aim at modular assembly structures that are not rigid and offer free flow assembly processes with respect to the assembly sequence (e.g. AUDI 2017).
To capture the complexity level of such assembly systems and of single assembly stations, numerous complexity measurement approaches exist in literature (see e.g. Bednar & Modrak 2014; ElMaraghy et al. 2012, Alkan et al. 2018). However, as shown in the next section, no complexity measure exists that captures the process-oriented complexity of the work of assembly teams. Many measures already focus on single assembly stations, but do not take the whole assembly flow and the interactions amongst the team members into account. If such a measure existed, work designers would be able to conduct studies on "optimized" levels of complexity for people working in assembly. Furthermore, the measure could be applied in early planning stages, if it was implemented into simulation models of assembly lines. To adapt existing measures for this purpose and to evaluate their usefulness, the paper is structured as follows:
Initially, existing complexity measures in the context of assembly systems are described and systematized. Subsequently, a measure to capture the complexity of the work of assembly teams is developed and explained. The implementation of the measure into a 3D simulation model[1] of a One-Piece-Flow assembly system is outlined and first results of the application of the measure are presented. Simulation studies are used to evaluate, whether the complexity measure is capable of capturing the complexity level adequately. The paper ends with a conclusion, critical discussion and outlines further research steps.

COMPLEXITY MEASURES FOR ASSEMBLY

Even though the term "complexity" is used in specific ways within different research disciplines, system theory is often employed to describe complexity (Hoeschen 2015). A complex system consists of many elements and is characterized by distinct interactions (Shalizi 2006, Simon 1996). Relevant literature on system theory outlines that certain characteristics of complex systems are: the number and type of elements and interactions, the change rate of the system structure and uncertainty of current and future system states as well as emergent events (multiple complexity definitions are presented e.g. in Hoeschen (2015), Blockhus (2010), Latos et al. (2018)).
A variety of approaches for measuring complexity in assembly can be found in literature. These approaches can be classified into entropic-information approaches, computational metrics/ heuristics and axiomatic design (Bednar & Modrak 2014; ElMaraghy et al. 2012). Subsequently, the main classification groups are elucidated and it is evaluated, if they can be adapted to develop the complexity measure for cooperative forms of work which can be applied already in early assembly planning phases:

[1] simulation with FlexSim (2018)

Entropic-information approach

The entropic-information approach was introduced by Shannon (1948) and quantifies complexity in bits as unit based on the assessment of uncertainty of information about system variables. Information-based measures provide comparable values for objective complexity. Frizelle & Woodcock (1995) first applied the approach of entropy to production. They measure the complexity of different production processes, machinery and layouts. This measure already focusses on assembly processes and summarizes the complexity level in one key figure. However, it has not yet been adapted to consider cooperative forms of work.

In terms of assembly, Ahmad et al. (2016) quantify the assembly complexity with respect to the mountability of the components and the assembly sequence using Shannon`s entropy. The measure is very suitable for regarding detailed assembly tasks and products, but does not mainly focus on the overall assembly flow.

Computational metrics/ heuristics

To quantify the manufacturing system complexity derived from the system configuration layout, graph theory can be applied (ElMaraghy et al. 2012). Approaches based on graph theory rather focus on structural complexity of a production layout. They do not explicitly take dynamic processes within the actual assembly flow into account. ElMaraghy and Urbanic (2003; 2004) consider quantity, diversity and content of information to quantify product, process and operational complexity with focus on the operating person. Even though the operator is in focus, objective complexity is measured by the methods mentioned. However, the ComplexityIndex developed by Mattsson et al. (2013; 2014) uses a questionnaire to measure subjective assembly complexity, which is perceived directly by the operator. Zeltzer et al. (2013) measure complexity on a workstation level for mixed model assembly systems and consider different complexity driving aspects (e.g. number of tools being used). All these approaches are very suitable for complexity assessments in existing manufacturing systems. In early planning phases, the required parameters for the measure are not available so that it is difficult to apply them when planning. Moreover, most of the measures rather focus on assembly stations than on the overall assembly process. Su et al. (2010) integrate a process complexity factor as well as a factor that determines the complexity induced by the product design into a method for predicting the frequency of assembly errors. During the planning stage, the empirically determined criteria for high and low complexity of the basic assembly complexity can also be applied to reduce assembly errors (Falck et al. 2014, 2017). This approach, which is employable in planning, focusses on single assembly tasks and not on the overall assembly flow for one product or the work process of an assembly team.

Axiomatic Design

An alternative approach is Axiomatic Design. Suh (1999) differentiates between time-dependent and time-independent complexity. Complexity is defined as the probability of design parameters meeting previously defined functional requirements, which can relate to production systems, products, processes or even work steps. These probabilities are quantified based on Shannon information entropy.

All in all, no complexity measure exists that captures the complexity of the work process of an assembly team. Some measures already focus on complexity of technical assembly processes. However, they do not focus on the interactions of the assembly team members during the work process. Moreover, some methods have been validated with case studies. None of the presented measures was validated via employing simulation studies. General measures based on information theory, which have not initially been designed for complexity measurement in assembly, are able to consider the whole assembly process and could be already applied during assembly planning. They only require probabilities of the system being in certain states as input. Therefore, it seems promising to adapt an entropic-information approach to assess the complexity of the work process of assembly teams.

DEVELOPMENT OF A COMPLEXITY MEASURE FOR THE WORK OF ASSEMBLY TEAMS

Complexity measures based on the entropic-information approach according to Shannon (1948) have been adapted to measure the complexity in manufacturing systems, especially the complexity of machines (ElMaraghy et al. 2012). The measure defines complexity as the entropy of the distribution of a machine α being in the states i (equation 1, in analogy to ElMaraghy et al. 2012):

$$C_{\mu} = -\sum_{S_i \in S} P(S_i) log_2 P(S_i) \qquad (1)$$

C_{μ} = statistical complexity
S_i = state i; $S_i \in S$
$P(S_i)$ = probability of machine α being in state i

The statistical complexity measure C_{μ} is expressed in bits and can be interpreted as the average amount of information that is stored in the process (ElMaraghy et al. 2012). The measure represents a degree of uncertainty. The single logarithmic function for one state is an inverted u-shape and has its maximum where the probability for the system – of either being in that specific state or not (bipartite system) – is 50 %. In this most uncertain case, it is most difficult to predict, whether the system will be in the regarded state or not when the attempt is made to predict the system status for the next time step.

This measure is adapted to measure the complexity of the work process of an individual operator in assembly. It is defined as entropy complexity measure of the probability distribution of employee e being in the states j (equation 2). In this article the states were defined as the following tasks which the operator may execute during the assembly process: assembling a product, pushing an assembly dolly, waiting since the assembly process is blocked and waiting due to overtaking maneuvers.

$$C_{\mu} = -\sum_{S_j \in S} P(S_j) log_2 P(S_j) \qquad (2)$$

C_{μ} = statistical complexity
S_j = state j; $S_j \in S$
$P(S_j)$ = probability of employee e being in state j

The initially presented measure to capture the complexity of one machine can be expanded to measure the complexity of a system that consists of components, e.g. several machines, by adding the dimension of components into the total formula, as shown in equation 3 (Deshmukh et al. 1998, Frizelle & Woodkock 1995, ElMaraghy et al. 2012):

$$C_{\mu} = -\sum_{k=1}^{K} \sum_{i=1}^{I} p_{ki} log_2 (p_{ki}) \quad (3)$$

C_{μ} = statistical complexity
i = state i; $i \in I$
k = component (machine) k; $k \in K$
p_{ki} = probability of machine k being in state i

This measure is adapted to measure the complexity of the work of an assembly team. Therefore, the existing measure is further developed by using it to consider the complexity level of the work processes for a group of employees. The individual complexity measure for one employee in equation 2 is expanded by the dimension of team members e that work in the assembly team with set E of employees in the team. The final complexity measure of the work process of an assembly team in a One-Piece-Flow system is presented in equation 4:

$$C_{\mu} = -\sum_{e=1}^{E} \sum_{j=1}^{J} p_{ej} log_2 (p_{ej}) \quad (4)$$

C_{μ} = statistical complexity
j = state (task) j; $j \in J$
e = employee e of the assembly team; $e \in E$
p_{ej} = probability of employee e being in state j

In order to evaluate the plausibility of this new defined complexity measure of the work process of an assembly team in a One-Piece-Flow system with simulation studies, the measure was implemented into a 3D simulation model.

IMPLEMENTATION OF THE MEASURE INTO THE ASSEMBLY LINE SIMULATION MODEL

A process-oriented and personnel-integrated 3D simulation model of a One-Piece-Flow assembly line was chosen as basis for the simulation studies. The model was originally developed for a washing machine assembly line of a manufacturer of white goods in Germany. The model is explained in detail in Latos et al. (2017). The processing logic of the single workstations is modeled by employing colored and stochastic Petri Nets. Three working modes (according to Arzet, 2005) of a One-Piece-Flow system were implemented in the Petri Net. In all modes, a work station is blocked as soon as an operator starts to assemble there. After completing the assembly step, the station is unblocked for the following person. In total, the line has 34 work stations.
In the first mode, the operators are allowed to overtake each other during the assembly process which is triggered by comparing their modelled performance indexes. The second mode does not allow overtaking. Finally, the third mode divides the line into predefined sections. The operators are assigned to single sections so that they only assemble a part of the product. In between the sections, buffers are integrated, which act as points where the operator hands over the product to the next colleague. The first working mode is used to evaluate the complexity measure. Overtaking maneuvers cause more interactions between the employees – these are characteristic for complex systems (c.f. section 2). For each task, the simulation uses proper MTM-UAS (Universal Analyzing System) task execution times. Because real assembly times vary, a beta distribution of the assembly times is implemented in the model. The simulation model calculates the assembly times in every station by multiplying the corresponding beta-distributed assembly time with the overall performance index and a station-wise performance index for every operator. During the simulation runs, all specific task types that each operator performs, along with their duration, are captured by the simulation model. In addition, general performance measures, such as output or throughput times, are tracked. The following task types were measured for all employees in the model: assembly, travelling (pushing of the assembly dolly) and waiting (this includes waiting because an assembly station is blocked and waiting because of an overtaking maneuver). It is important to remember that, for example, the assembly time is also influenced by the operator's abilities (modelled in performance indexes) or by disturbances in the assembly process. The specific task times are then converted into percentage ratios with respect to a whole shift.
The complexity measure was programmed as a further performance measure of a shift. Initially, the complexity measure is calculated for every single operator according to equation 2. Then, the single measures are summed up according to equation 4 to calculate the complexity measure of the whole assembly team. The complexity measure is saved for every replication (repeated simulation run for a specific scenario). The single complexity measures are used to calculate statistical key figures, such as the mean, standard deviations and confidence intervals. The statistical evaluation of the plausibility of the complexity measure depends on the number of replications. Therefore, a sufficient high number of replications per scenario has to be determined.

EVALUATION OF THE COMPLEXITY MEASURE IN SIMULATION STUDIES

In order to evaluate the plausibility of the complexity measure, the characteristics of complex systems (section 2) are considered: it was stated that the level of complexity rises with the number of elements and the number of interactions between the elements in a system. Since the measure captures the complexity of the system "assembly team", the number of elements is the number of operators. These operators have interactions during the assembly processes. For example, stations are blocked by other operators or overtaking maneuvers due to the performance indexes of the operators take place. According to equation 4, the complexity measure should increase, if the number of persons in the team increases. Therefore, the following proposition is formulated:
Proposition 1: *The higher the number of operators in the assembly team, the higher the level of complexity of the work process of the assembly team will be.*
If the number of persons in the system rises, the number of interactions will also increase. With respect to the phenomenon of emergent events, it is assumed that more dynamic interaction processes will take place so that

unexpected events will occur. Therefore, the variation in the output and complexity measures should increase, as the number of persons in the system increases.

Proposition 2: *The higher the number of persons in a system, the higher the variation of the complexity measure will be. The same proposition is made for the output measure.*

Finally, it was outlined in section 2 that complexity is also caused by the type and difference between system elements. The operators, as system elements, are modelled with different performance indexes. It is assumed that the individual complexity levels differ between operators with high and low complexity levels. This is assumed because the performance level affects, whether overtaking maneuvers will take place. Therefore, this has an impact on the ratio of assembly time.

Proposition 3: *Operators with low performance indexes have a different individual complexity level in comparison to operators with high performance indexes.*

To evaluate the complexity measure and test the propositions with simulation studies, the number of replications per simulation run has to be set. Therefore, a power analysis for a pairwise comparison between one scenario with one of the two nearest other scenarios was conducted in GPower (Faul et al., 2009). The following parameters were used as input parameters for the power analysis for an independent sample *t*-test: A very small standardized effect size Cohen's $d = 0.20$ (cf. Cohen, 1988), a two-sided significance level $\alpha = 0.05$, and a high target level of power of 0.95 (see Cohen, 1992). The power analysis yielded a required sample size of 651 replications per scenario. Every replication represents one shift simulation. Consequently, a simulation study was set up that repeatedly (651 times) estimated the complexity and the output for 25 scenarios (one to 25 employees). Nonlinear regression models with the number of employees as predictor and output, respectively complexity, as criterion were fitted to the data. This approach was chosen, since it enables a comparative evaluation of all scenarios. A second-order polynomial was used to fit both distributions (see Figure 1 and Figure 2). Both regressions explain substantial variance in their respective criterion: $R^2 = .99$ ($p < .001$) for output as well as for complexity (analyses with IBM SPSS Statistics 23). The left-hand sides of the figures show the mean and standard deviation (for better visualization, two standard deviations are displayed) of the output and complexity measure for each number of employees; the right-hand sides of the figures show the respective curves fitted to these means as well as the equations.

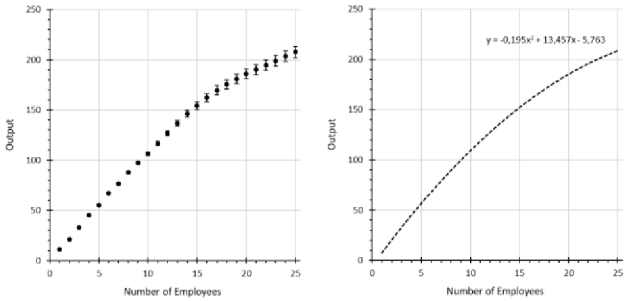


Figure 1: Output per shift depending on the number of employees in the assembly system (651 replications per scenario; with overtaking)

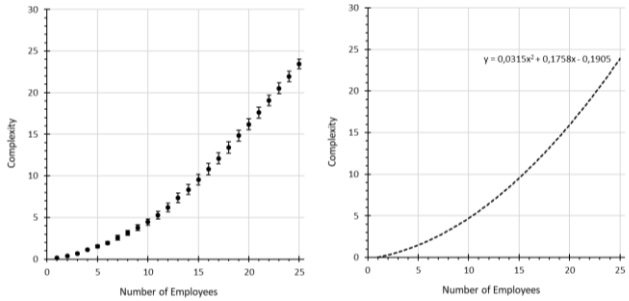


Figure 2: Complexity measure of the work process of the assembly team depending on the number of team members in the assembly system (651 replications per scenario; with overtaking)

The output curve flattens, which can also be seen in the polynomial in Figure 1. This flattening occurs, if too many employees are in the assembly system and too many blocking events occur. Consequently, the ratio of assembly time diminishes. This phenomenon can also be seen in the complexity curve (Figure 2), whereas the diminishing assembly ratios cause that the complexity measure increases in a quadratic manner. This is confirmed by the polynomial. *Therefore, proposition 1 is retained at this point.*

From a descriptive point of view, the standard deviations, which are represented in two standard deviations in Figure 1 and Figure 2, seem to increase in both output and complexity curves. Yet, further empirical investigation is needed to confirm this. *So far, proposition 2 is retained.*

On the basis of these findings, a nonlinear regression model was developed and fitted to the data with the complexity measure as predictor and the output per shift as criterion. A fourth-order polynomial was used to fit the relationship. Figure 3 illustrates the result and presents the polynomial. The higher the complexity, the higher the output becomes. However, the output flattens, whilst the complexity measure increases. Figure 4 summarizes the findings as 3D plots.

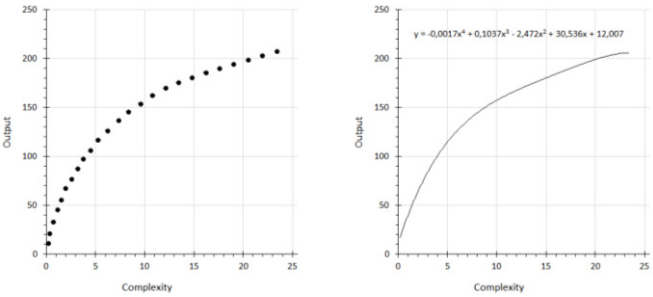


Figure 3: Output per shift depending on the complexity measure of the work process of the assembly team (651 replications per scenario; with overtaking)

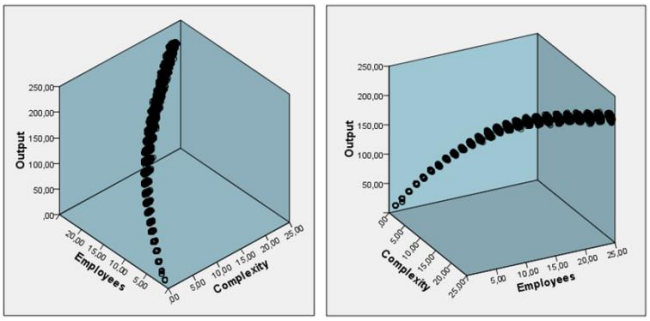


Figure 4: Output per shift, complexity measure of the work process of the assembly team and the number of employees in the assembly team as 3D plots from two perspectives (651 replications per scenario; with overtaking)

Furthermore, the complexity level, that arises for the employee with the lowest and highest performance index in every simulation replication, was also estimated with respect to the number of employees in the assembly team and is depicted in Figure 5. In this case, no overtaking during the assembly process was allowed. This setting was chosen, since it was assumed that this working mode of a One-Piece-Flow system will mostly affect the ratios of assembly time when comparing the two employees. It is noticeable that the complexity level of the person with the highest performance index increases rapidly and remains on a high level whilst the number of employees increases. The curve represents that this person spends a considerable time of the shift waiting for employees with lower performance indexes in the next assembly station to finish work there. The faster employee is blocked, since no overtaking is allowed. In contrast, the person with the lowest performance level exhibits a relatively low complexity level, which is retained up to 20 employees in the system. From this point on the system is too full so that this complexity level also increases. This curve progression can be explained in such an extent that all other persons will queue up behind this person. However, since no overtaking is allowed, the person will be able to continue the assembly process at all next stations. Of course, this constellation could effectuate social tensions amongst the employees so that the productivity of the person with the lowest performance level could also decline. In turn, the individual complexity measure of this person would increase. *The findings indicate that proposition 3 can be retained at this point.*

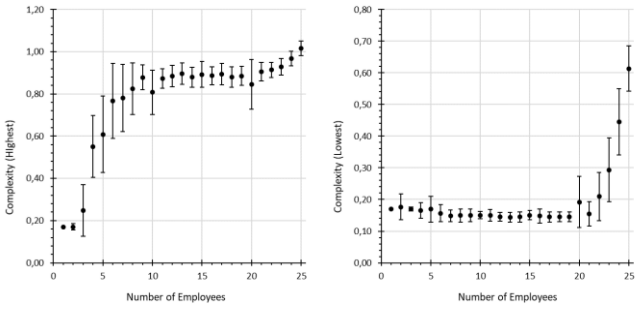


Figure 5: Individual complexity measure of the person with the highest performance index (left) and the person with the lowest performance index (right) depending on the number of team members in the assembly system (651 replications per scenario; without overtaking)

DISCUSSION AND FUTURE RESEARCH

In this article, a new measure was described to capture the complexity of the work of assembly teams in One-Piece-Flow systems. The measure focusses fully on cooperative forms of work and on assembly processes. It does capture different activities during the assembly process and calculates the complexity level with respect to a whole shift. The complexity level is represented by one key figure in the unit bits. This makes it easy to compare different scenario settings. Further descriptions and interpretation guidelines would facilitate the application for practitioners.

The plausibility of the measure was evaluated with simulation studies. However, the evaluation procedure should be complemented with ongoing statistical analyses. In this context, the propositions may be formulated as hypotheses and should be evaluated with appropriate statistical test methods. Further research steps will include the comparison of different working modes of a One-Piece-Flow system (overtaking; no overtaking; group-wise assembly). Moreover, a combination of different complexity assessment methods with respect to the level of detail (assembly process vs. assembly station considerations) within different planning phases should be merged into an holistic approach.

Also, future studies will have to show, whether this measure is transferrable to other assembly systems. In this context, it must be stated that the interpretation of the complexity measure always has to be conducted with a differentiated view: the interpretation depends on the task types that are predefined for the complexity measure. In this case, the two task types (travelling and waiting), beside the assembly task, were non-productive task types. When these tasks had larger ratios, and therefore diminished the assembly ratio towards 50 %, the complexity key figure increased. If other important task types were to be defined, such as “quality control”, the measure should be reinterpreted. This leads to the final remark that complexity can have negative as well as positive effects in a work system. On the one hand, it may cause overstraining work conditions for employees; on the other hand, it may also make job contents more challenging and interesting.

CONCLUSION

This paper presents the development of a complexity measure for the work process of assembly teams in One-Piece-Flow assembly systems. The measure is developed on the basis of Shannon’s entropy approach. Furthermore, the measure is implemented into a 3D simulation model of a One-Piece-Flow system. Simulation studies are conducted to evaluate the complexity measure. The results show that the measure is capable of capturing the complexity of the work process of an assembly team. The measure can be used in the context of work design to provide work systems with a good tradeoff between output and complexity key figures. Furthermore, the measure is used to present results of an initial practical application of the measure. Field studies will have to show whether this measure can be integrated into useful preventive analysis procedures in early stages of work system planning.

ACKNOWLEDGEMENT

The research and development project "TransWork" is funded by the German Federal Ministry of Education and Research (BMBF) within the program "Innovationen für die Produktion, Dienstleistung und Arbeit von morgen" according to Grant No. 02L15A162 supervised by the Project Management Agency of Karlsruher Institute of Technology (PTKA). The authors are responsible for the content of this publication. The authors would like to express their gratitude for this support.

REFERENCES

Ahmad, M.; Alkan, B.; Ahmad, B.; Vera, D.; Harrison, R.; Meredith, J.; and Bindel, A. 2016. "The Use of a Complexity Model to Facilitate in the Selection of a Fuel Cell Assembly Sequence". In *Procedia CIRP*, 44, 169-174.

Alkan, B.; Vera, D. A.; Ahmad M.; Ahmad, B.; Harrison, R. 2018. "Complexity in manufacturing systems and its measures: a literature review" In *European J. of Industrial Engineering*, 12 (1), 116-150

Arzet, H. 2005. "Grundlagen des One-piece-flow". *Guideline for planning and realizing operator-based production systems (in German)*. Rhombos, Berlin, Germany.

AUDI AG. 2017. "Dialoge smart factory". *Dialoge-Magazin 1*.10-15.

Bednar, S. and Modrak, V. 2014. "Mass Customization and its Impact on Assembly Process' Complexity". In *International Journal for Quality Research*, 8(3), 417-430.

Blockus, M. O. 2010 "Komplexität in Dienstleistungsunternehmen: Komplexitätsformen, Kosten- und Nutzenwirkungen, empirische Befunde und Managementimplikationen" (in German), Ph.D. dissertation, Universität Basel, Switzerland, 2010, 1st ed., Gabler Research, Wiesbaden, Germany.

Cohen, J. (1988). "Statistical power analysis for the behavioral sciences", 2nd ed., Taylor & Francis, New York

Cohen, J. 1992. "A power primer". In *Psychological Bulletin*, 112 (1), 155–159.

Deshmukh, A.V.; Talavage, J.J.; and Barash, M.M. 1998. "Complexity in manufacturing systems, Part 1: Analysis of static complexity". In *IIE Transactions*, 30 (7), 645–655.

ElMaraghy, W.H. and Urbanic, R.J. 2003. "Modelling of Manufacturing Systems Complexity". In *CIRP Annals - Manufacturing Technology*, 52 (1), 363-366.

ElMaraghy, W.; ElMaraghy, H.; Urbanic, R. J. 2004: Assessment of Manufacturing Operational Complexity. In: CIRP Annals - Manufacturing Technology, 53 (1), S. 401–406

ElMaraghy, W.; ElMaraghy, H.; Tomiyama, T.; and Monostori, L. 2012. "Complexity in engineering design and manufacturing" In *CIRP Annals – Manufacturing Technology*, 61 (2), 793-814.

Falck, A.-C.; Örtengren, R.; and Rosenqvist, M. 2014. "Assembly failures and action cost in relation to complexity level and assembly ergonomics in manual assembly (part 2)". In *International Journal of Industrial Ergonomics*, 44 (3), 455-459.

Falck, A.-C.; Örtengren, R.; Rosenqvist, M.; and Söderberg, R. 2017. "Proactive assessment of basic manual assembly: development of a tool to predict and control operator-induced quality errors". In *International Journal of Production Research*, 55 (15), 4248-4260.

Faul, F.; Erdfelder, E.; Buchner, A.; and Lang, A.G. 2009 "Statistical power analyses using G*Power 3.1: Tests for correlation and regression analyses". In *Behavior Research Methods*, 41, 1149-1160.

Frizelle, G. and Woodcock, E. 1995. "Measuring complexity as an aid to developing operational strategy". In *International Journal of Operations & Production Management*, 15 (5), 26-39.

Hoeschen, A. 2015. "Complexity-based distribution of value added in multinational production systems" (Thesis in German) Ph.D. dissertation, RWTH Aachen University, Germany.

Latos, B.A.; Kalantar, P.; Przybysz, P.M.; and Mütze-Niewöhner, S. 2017 "Simulation of a Flexible and Adaptable One-Piece-Flow Assembly Line Based on a Process Flow of Colored and Timed Petri Nets". In *Proceedings 2017 European Simulation And Modelling Conference*, Lisbon, Portugal, 162-166.

Latos, B. A.; Harlacher, M.; Burgert, F.; Nitsch, V.; Przybysz, P.; Mütze-Niewöhner, S. 2018 "Complexity Drivers in Digitalized Work Systems: Implications for Cooperative Forms of Work". In *Advances in Science, Technology and Engineering Systems Journal,* 3(5), 166-185.

Mattsson, S. 2013. "What is perceived as complex in final assembly? To define, measure and manage production complexity". Diss. Chalmers University of Technology, Gothenburg, Sweden.

Mattsson, S.; Karlsson, M.; Gullander, P.; van Landeghem, H.; Zeltzer, L.; Limère, V.; Aghezzaf, E.H.; Fasth, Å.; and Stahre, J. 2014. "Comparing quantifiable methods to measure complexity in assembly". In *International Journal of Manufacturing Research*, 9 (1), 112-130.

Shannon, C.E. 1948. "A Mathematical Theory of Communication". In *The Bell System Technical Journal*, 27 (3), 379-423.

Schuh, G.; Riesener, M.; Kantelberg, j.; and Steireif, N. 2017. "Transmission of software-related agile mechanisms of action towards product development processes for technical products". In *IEEE International Conference on Industrial Engineering and Engineering Management*, Singapore, Singapore.

Shalizi, C.R. 2006. "Methods and Techniques of Complex Systems Science". In *T.S.Deisboeck, J.Y.Kresh (Ed.) Complex systems science in biomedicine*, Springer, New York, USA, 49-131.

Simon, H.A. 1996. *The sciences of the artificial*, 3rd ed., MIT Press Cambridge, USA.

Spath, D.; Müller, R.; and Reinhart, G. 2013. *Future Assembly Systems. Economic, Changeable and Reconfigurable (in German)*. Fraunhofer, Stuttgart, Germany.

Su, Q.; Liu, L.; and Whitney, D.E. 2010. "A Systematic Study of the Prediction Model for Operator-Induced Assembly Defects Based on Assembly Complexity Factors". In *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 40 (1), 107-120.

Suh, N.P. 1999. "A Theory of Complexity, Periodicity and the Design Axioms". In *Research in Engineering Design*, 11 (2), 116-132.

Zeltzer, L.; Limère, V.; van Landeghem, H.; Aghezzaf, E.-H.; Stahre, J. 2013 "Measuring complexity in mixed-model assembly workstations" In *International Journal of Production Research*, 51 (15), 4630-4643.

WEB REFERENCES

FlexSim Software Products, Inc. (FSP). 2018. FlexSim 2018 manual, URL: https://www.flexsim

BIOGRAPHIES

BENEDIKT A. LATOS was born in Baden-Baden, Germany and went to the University of Bath in England and RWTH Aachen University in Germany, where he studied Business Administration & Engineering: Mechanical Engineering with specialization in production techniques and obtained his M.Sc. degree at RWTH Aachen University with distinction in 2016. Currently, he works as research assistant and doctoral student in the field of Work Organization at the Institute of Industrial Engineering an Ergonomics of RWTH Aachen University. His lecturing units relate to discrete event simulation as well as timed and colored Petri Nets. His research topics concern the transformation of work through digitalization, complexity management in assembly and assembly system simulation.

PEYMAN KALANTAR was born in Shiraz, Iran. He completed his B.Sc degree in Mechanical Engineering Fluid Dynamics at Yasouj University in 2012 in Iran. He has experience in working in machine industry in Iran (2012-2013) and also internationally in the light industry-jewelry section, MKS Jewelry Co, LTD in Bangkok, Thailand as Acting Production Manager (2013-2016). He is currently studying Management and Engineering in Computer Aided Mechanical Engineering at RWTH Aachen University in Germany and is working as research student assistant at the Institute of Industrial Engineering and Ergonomics of RWTH Aachen University. His research interest is in simulation of discrete event systems, sustainable operations and complexity management in assembly systems.

FLORENS L. BURGERT was born in Herdecke, Germany. He obtained his B.Sc. in Business Administration & Engineering: Mechanical Engineering with specialization in production techniques at RWTH Aachen University in Germany in 2017. Currently, he is pursuing his Master's degree. He is working as research student assistant at the Institute of Industrial Engineering and Ergonomics of RWTH Aachen University. His research interests focus on complexity management in production and assembly and digitalization in the context of production systems.

MATTHIAS G. AREND was born in Filderstadt, Germany. He completed his B.Sc. degree in Psychology at Chemnitz University of Technology in 2017. Currently, he is completing his M.Sc. degree in Psychology at RWTH Aachen University in Germany and is as well working as research student research assistant at the Institute of Industrial Engineering and Ergonomics of RWTH Aachen University. His research interests include the development of eco-driving interfaces from the perspective of behavioral economics, quantitative methods in psychology and complexity management in production and assembly and digitalization in the context of production systems.

VERENA NITSCH is Professor and Head of the Institute of Industrial Engineering and Ergonomics at RWTH Aachen University, Germany. She completed her Bachelor's degree in Applied Psychology at Charles Sturt University in Australia and the University of Central Lancashire in Great Britain, where she graduated with distinction in 2006. She received her Master's degree in Organizational Psychology from Manchester Business School in 2007. In 2008, she became a research assistant at the Human Factors Institute at the Bundeswehr University Munich, where she received an engineering doctorate with summa cum laude in 2012. In 2013, Verena Nitsch was appointed Junior Professor of Cognitive Ergonomics at the Bundeswehr University Munich, where she also led the Human Factors Institute from 2016-2018. Since 2018, she is Professor and Head of the Institute of Industrial Engineering and Ergonomics at RWTH Aachen University. She is active as an expert and consultant in numerous committees and advisory boards. Her research interests include human-technology interaction in semi-automated and highly automated work systems as well as user interface design.

PHILIPP M. PRZYBYSZ was born in Wloclawek, Poland. He completed M.Sc. degrees in Work and Organization Psychology (in 2010) and in International Business: Strategy and Innovation (in 2011) at Maastricht University. His doctoral studies concerned the impact of age diversity on the work of innovation teams. He obtained his Dr. rer. nat. degree at Ernst-Moritz-Arndt-Universität Greifswald, Germany in 2016. Currently, he works as deputy Head of Department in the field of Work Organization at the Institute of Industrial Engineering and Ergonomics at RWTH Aachen University, Germany. His research topics relate to innovation management, the transformation of work through digitalization and complexity management.

SUSANNE MÜTZE-NIEWÖHNER was born in Solingen, Germany. After completing her studies in Mechanical Engineering at RWTH Aachen University she worked as research assistant at the Institute of Industrial Engineering and Ergonomics of RWTH Aachen University. Her doctoral thesis (published in 2004) concerned the prospective design of work tasks in production in the context of concurrent engineering processes. Currently, she is Professor for organizational design and development at the faculty of mechanical engineering of RWTH Aachen University and she works as Head of Department in the field of Work Organization at the Institute of Industrial Engineering and Ergonomics at RWTH Aachen University, Germany. Her research topics relate the analysis, evaluation and design of (socio-technical) work systems in the context of current demands and with special consideration of human-oriented criteria; investigation and development of participatory, team-based organizational structures and processes.

INTEGRATED EPQ AND PERIODIC CONDITION-BASED MAINTENANCE

Alp Darendeliler
Dieter Claeys
El-Houssaine Aghezzaf
Department of Industrial Systems Engineering and Product Design
Ghent University, Belgium
and
Industrial Systems Engineering (ISyE), Flanders Make

Abdelhakim Khatab
Industrial Engineering and Production Laboratory, Lorraine University
Metz, France

KEYWORDS
Condition-based maintenance, Economic Production Quantity.

ABSTRACT

In this paper, a stochastically deteriorating production system is studied under condition-based maintenance. Periodic monitoring is carried out to observe the degradation level of the system. If the degradation level exceeds failure threshold, nonconforming items are produced and a high corrective maintenance cost is incurred. Preventive maintenance actions are performed to reduce the possibility of failures. By considering inspection interval, preventive maintenance level and lot-size as decision variables, an integrated model is developed to minimize long-run average cost rate consisting of inspection costs, maintenance cost, cost of producing nonconforming items, inventory holding cost and setup costs. An illustrative example is presented to analyze the model.

INTRODUCTION

Condition-based maintenance (CBM) is an approach which recommends maintenance actions according to the current status of the production system through condition monitoring (Jardine et al., 2006). Due to the development of the sensor technologies, the current degradation level of the manufacturing systems can be monitored. The other maintenance approaches are classified as unplanned (breakdown) maintenance and time-based (planned) preventive maintenance (Martin, 1994). Unplanned maintenance takes place when a failure occurs. In this case, unexpected failures can interrupt the production plan and cause lost sales. Thus, it is an inefficient approach. Under time-based maintenance, periodic preventive maintenance actions are performed at certain points in time to reduce the possibility of failures. The current health status of the system is not taken into account. Therefore, unnecessary preventive maintenance actions could be done when there is remaining health of the system. CBM can eliminate these unnecessary maintenance actions by taking maintenance actions according to the state of the system through conditional monitoring.

The degradation of the components can either be monitored via continuously, periodically or non-periodically. Continuous monitoring gives real-time data about the degradation level of the system. However, in some production systems like pipelines buried underground in oil and gas industries, performing continuous monitoring is not applicable (Alaswad and Xiang 2017). These systems can be monitored at certain points in time to assess the degradation level of the components. For the cases where the inspection costs are high, an economic inspection policy needs to be found to reduce the overall cost.

Ben-Daya and Makhdoum (1998) consider an integrated production and quality model for different inspection policies. The deterioration process is modeled by a hazard rate function. Another integrated optimization of lot-sizing and preventive maintenance level, is developed by Ben-Daya (2002). The age-based maintenance policy is applied. In that model, optimal preventive maintenance level and inspection intervals are found to minimize the total cost of production and maintenance. Chen (2013) constructed a model to optimize EPQ and preventive maintenance level under Weibull shock model. Jafari and Makis (2015) studied optimal lot-sizing and preventive maintenance policy where the deterioration is modeled by the proportion hazards model.

In the literature, joint optimization EPQ and CBM has been studied. Peng and Van Houtum (2016) proposes a model to jointly optimize the total cost rate associated with the production lot-sizing and condition-based maintenance. In their model, they assume that the degradation process is continuously monitored and it is modeled as a continuous time and continuous state stochastic process. Khatab et al. (2017) develop a model to minimize long-run average cost rate of total production and maintenance costs by finding optimal values of preventive maintenance level and inspection intervals. Gamma process is used for modeling the degradation. However, the lot-size is not optimized. Cheng et. al (2017) develop a model for joint optimization of lot-sizing and CBM for multiple dependent items that are economically dependent. Inspections are carried out at the end of the production lots. They use simulation and a genetic algorithm to find the optimal values of lot-size and preventive maintenance level.

In the literature, periodic monitoring has not been considered for the optimization of lot-sizing and CBM. In this study, an integrated model is developed to optimize production and maintenance costs simultaneously under condition-based maintenance where the system is monitored periodically to observe the degradation level. The degradation of the component is modeled as a stationary Gamma Process which

fits well for modeling temporal variability of deterioration (Van Noortwijk, 2009). A considerable cost is charged for each inspection so it is necessary to determine an optimal inspection interval in order to minimize the total cost rate. Nonconforming items are produced in case the system degradation level exceeds the failure threshold. Additional cost is incurred due to the production of nonconforming items. The setup cost, inventory holding cost, preventive and corrective maintenance cost, inspection cost and cost of producing nonconforming items are considered for the minimization of the total long-run average cost rate.

MODEL DESCRIPTION

A single component and single item production system is considered in this paper. The deterioration level of the component can only be observed upon completion of the production lot and it is modeled as a stationary Gamma process $X(t)$, which is a monotonically increasing function. The system is out of control when the degradation level exceeds failure threshold X_f. In this case, nonconforming items are produced and additional cost is incurred for the production of each nonconforming item. We assume that the manufacturing system produces at a constant production rate p and a constant demand rate d.

In the model, the time length of production for a lot t_0 ($Q = (p-d)t_0$), the preventive maintenance threshold X_p and the inspection interval τ are the decision variables that should be determined to minimize the long-run average cost rate $C(\tau, t_0, X_p)$. The frequency of inspection τ, is an integer multiple of the time length of production t_0. Inspections are carried out at times $\tau, 2\tau, \ldots$, when the production of a lot ends. Deterioration occurs during the production time so its level remains same during the idle time of the lots which starts after the inventory level $I(t)$ reaches $(p-d)t_0$. Initial inventory level is assumed to be zero. Preventive and corrective maintenance actions take a fixed amount of time l and carried out during the idle times in order not to interrupt the production. It is assumed that after preventive and corrective maintenance, the system is "as good as new".

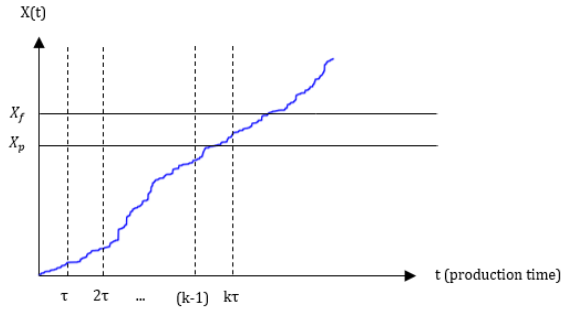


Figure 1. Sample degradation path crossing preventive maintenance level

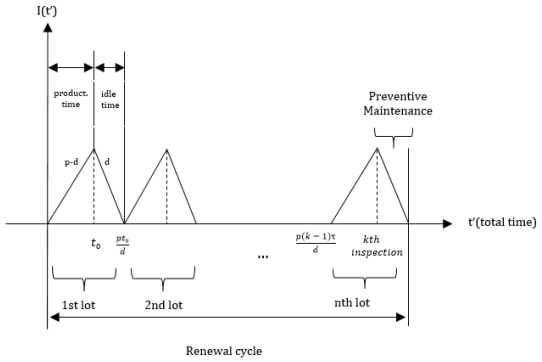


Figure 2. Inventory level in one cycle in case of preventive maintenance

A sample degradation path with respect to production time, is presented in Figure 1. X_p is the preventive maintenance threshold; if the degradation level is observed to be higher, then preventive maintenance is performed. In Figure 1, since $X(t)$ is between X_p and X_f at time $k\tau$, preventive maintenance is performed right after the k^{th} inspection to avoid failure and costly corrective maintenance. In this graph, only production time is considered because the degradation level remains unaltered during the idle times. The corresponding inventory level with respect to the total time t', is shown in Figure 2. The setup cost C_s, is incurred for each production lot. Time t' consists of production and idle times.

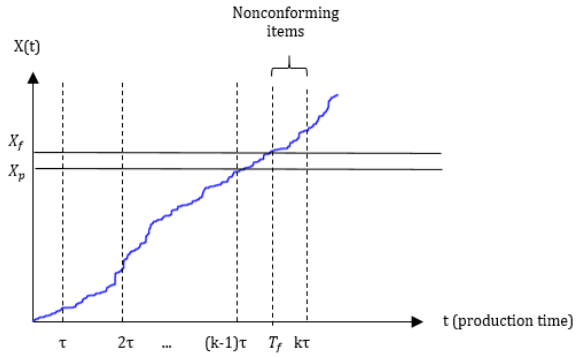


Figure 3. Sample degradation path crossing failure threshold level

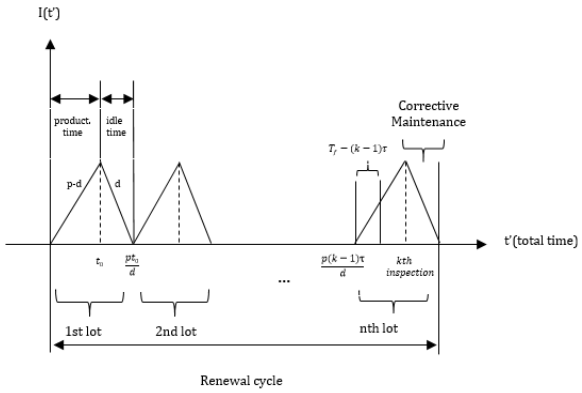


Figure 4. Inventory level in one cycle in case of corrective maintenance

The path in Figure 3 illustrates the case where corrective maintenance actions are to be performed at the end of the kth inspection. The deterioration does not reach the level X_p before the $(k-1)^{th}$ inspection. In the production of the lot corresponding to the k^{th} inspection, it exceeds the failure threshold. Thus, nonconforming items are produced which incur additional costs. Figure 4 shows the corresponding graph of inventory level over total time.

Notations

i	index of inspection intervals
τ	inspection interval
p	constant production rate
d	constant demand rate
t_0	production time for a lot
Q	lot-size
l	constant duration of predictive and corrective maintenance ($l \leq \frac{pt_0}{d} - t_0$)
$X(t)$	degradation level of the production system at time t
$F(x, t-s)$	cumulative distribution function of $X(t) - X(s)$
$f(x, t-s)$	probability density function of $X(t) - X(s)$
X_p	predetermined threshold level for preventive maintenance
X_f	failure threshold level
C_h	inventory holding cost per unit of time
C_I	cost for one inspection
C_s	setup cost per lot
C_{nc}	cost of producing one nonconforming unit
C_p	cost of the predictive maintenance action
C_c	cost of the corrective maintenance action
T_f	first passage time to the failure threshold
α	percentage of producing nonconforming items when the degradation level is above X_f
$E[S_c]$	expected setup cost per cycle
$E[H_c]$	expected inventory holding cost per cycle
$E[M_c]$	expected maintenance cost per cycle
$E[NC_c]$	expected cost of producing nonconforming items per cycle
$E[T]$	expected cycle length
$E[C]$	expected total cost in one cycle
$\mathrm{C}(\tau, t_0, X_p)$	long-run average total cost per unit of time

FORMULATION OF THE OPTIMIZATION MODEL

The degradation of the component is modeled as a stationary Gamma process with shape and scale parameters α and β respectively. It suits well with condition-based maintenance models where inspections are carried out in discrete time points (Van Noortwijk, 2009). The density function of the deterioration of $X(t_1) - X(t_2)$ between times t_1 and t_2 is as follows:

$$f(x, t_1 - t_2) = \frac{x^{((t_1-t_2)k-1)} \exp(-\frac{x}{\theta})}{\Gamma((t_1 - t_2)\, k)\theta^{(t_1-t_2)k)}} \quad (1)$$

and the cumulative density function is computed by the equation,

$$F(x, t_1 - t_2) = \frac{\Gamma((t_1 - t_2)k,\ \frac{x}{\theta})}{\Gamma((t_1 - t_2)k)} \quad (2)$$

where $\Gamma\big((t_1 - t_2)\, k\big)$ is the gamma function and $\Gamma((t_1 - t_2)\, k, \frac{x}{\theta})$ is the lower incomplete gamma function. The cumulative density function of the first passage time to failure threshold T_f, is

$$G(t) = P\{T_f \leq t\} = P\{X(t) > X_f\} = \overline{F}(X_f, t) \quad (3)$$

The density function of the first passage time to the failure threshold is $g(t) = \frac{d}{dt}\overline{F}(X_f, t)$. Khatab et al. (2017) express this density function as

$$g(t) = \frac{k}{\Gamma(kt)} \int_{\frac{Xf}{\theta}}^{\infty} [\ln(u) - \Psi(kt)] u^{kt-1} \times \exp(-u)\, du \quad (4)$$

where $\Psi(u) = dln(\Gamma(u))/du$ is the digamma function.

After either preventive or corrective maintenance, the degradation level becomes zero. Figure 2 shows an example of a renewal cycle. After the maintenance action is completed and the inventory level becomes zero, the renewal cycle restarts. The renewal reward theorem is used to compute the long-run average cost rate by dividing the average total accumulated cost in a renewal cycle by the average cycle length.

The expected cycle length is calculated by finding probability of the event that degradation level is lower than X_p before $(i-1)^{th}$ inspection and it exceeds X_p between $(i-1)^{th}$ and i^{th} inspections (Figure 1). It is given by

$$E[T] = \sum_{i=1}^{\infty} \frac{i\tau p}{d} \int_0^{X_p} f(x, (i-1)\tau) \bar{F}(X_p - x, \tau) dx \quad (5)$$

where $(\tau p)/d$ is the total time length between two consecutive inspections.

The expected inventory holding cost per cycle is computed by

$$E[H_c] = C_h E[T] \frac{(p-d)t_0}{2} \quad (6)$$

The expected sum of inspection and maintenance cost per cycle is,

$$E[M_c] = C_I \frac{E[T]}{\left(\frac{\tau p}{d}\right)} + \sum_{i=1}^{\infty} \int_0^{X_p} f(x,(i-1)\tau)(C_p \bar{F}(X_p - x, \tau) + (C_c - C_p)\bar{F}(X_f - x, \tau)dx \quad (7)$$

where the probability of performing maintenance right after the i^{th} inspection is $\int_0^{X_p} f(x,(i-1)\tau)\bar{F}(X_p - x, \tau)dx$. The probability of performing corrective maintenance in a cycle is as follows:

$$P\{E_i^c\} = P\{((X(i\tau) \geq X_f)\ \&\ X((i-1)\tau) < X_p)\} = \int_0^{X_p} f(x,(i-1)\tau)\bar{F}(X_f - x, \tau)dx \quad (8)$$

The expected setup cost per cycle is given by

$$E[S_c] = C_s \frac{E[T]}{(pt_0/d)} \quad (9)$$

where (pt_0/d) is the time length of one production lot.

If the degradation level $X(i\tau)$ is observed to be x at any inspection, then probability density function of the remaining time to failure T_f given that $X(i\tau) = x$, is as follows:

$$g(x,t) = \frac{d}{dt} P\{T_f \leq t | X(0) = x\} = \frac{d}{dt}\bar{F}(X_f - x, t) \quad (10)$$

The conditional density function of T_f given that E_i^c can be expressed as

$$f_{T_f}((i-1)\tau + t | E_i^c) = \frac{\int_0^{X_p} f(x,(i-1)\tau)g(x,t)dx}{P\{E_i^c\}} \quad (11)$$

By using the above probability, the expected cost of producing nonconforming items given the event E_i^c can be calculated as

$$E[NC_c | E_i^c] = C_{nc}\alpha p \int_0^{\tau} (\tau - t) \frac{\int_0^{X_p} f(x,(i-1)\tau)g(x,t)dx}{P\{E_i^c\}} dt \quad (12)$$

where α is the percentage of the produced nonconforming items and C_{nc} is the cost of producing a nonconforming item. After the degradation level reaches X_f, nonconforming items are produced up to the time of i^{th} inspection. An example path is shown in Figure 3. By multiplying the above equation by $P\{E_i^c\}$ for each inspection i and summing over all probabilities, the expected cost of producing nonconforming items is computed. It is expressed as

$$E[NC_c] = \sum_{i=1}^{\infty} C_{nc}\alpha p \int_0^{\tau}\int_0^{X_p} (\tau - t) \times f(x,(i-1)\tau)g(x,t)dxdt \quad (13)$$

For each value of the inspection interval $(\tau = nt_0)$, the optimization problem is to minimize the long-run expected cost rate $C(\tau, t_0, X_p)$ by finding the optimal values of t_0, X_p subject to the constraint that idle time period $(p-d)t_0/d$, is longer than the time length of the predictive and corrective maintenance actions. Otherwise, shortages occur. The model is as follows:

$$minimize \quad C(\tau, t_0, X_p)$$

$$subject\ to$$

$$\frac{(p-d)t_0}{d} \geq l \quad (14)$$

$$X_p \geq 0 \quad (15)$$

$$X_f - X_p \geq 0 \quad (16)$$

where the long-run average cost rate is,

$$C(\tau, t_0, X_p) = \frac{E[C]}{E[T]} = \frac{E[H_c] + E[M_c] + E[S_c] + E[NC_c]}{E[T]} \quad (17)$$

and the optimal objective function value of the above optimization problem is $C(\tau, t_0^*, X_p^*)$.

The optimal long-run average cost rate of $C(\tau, t_0, X_p)$, is found by solving the optimization problem for each value of $\tau = t_0, 2t_0, \ldots, Mt_0$ where M is a sufficiently big integer. Thus, the optimal value is,

$$C(\tau,^* t_0^*, X_p^*) = \min_{\tau = t_0, 2t_0, \ldots, Mt_0} C(\tau, t_0^*, X_p^*) \quad (18)$$

The objective function of the optimization problem is differentiable. Examples of different data sets show that the objective function is not convex so the local minimums might not be the global minimum. The Frank-Wolfe algorithm is used to solve this problem (Hillier and Lieberman 2001). The algorithm uses the linear approximation of the nonlinear objective function that are obtained by the first-order Taylor

series expansion. Different initial points are chosen to find the local minimum with the smallest objective function value.

AN ILLUSTRATIVE EXAMPLE

A computational result of the model is presented in this section. The deterioration of the system $X(t)$ is modeled as a stationary Gamma process. The shape and scale parameters are $k = 1.2$ and $\theta = 0.8$ respectively. Inspection interval τ, is set as the integer multiple of t_0. Thus, inspections are carried out right after the completion of the production. The inspection, preventive maintenance and corrective maintenance costs are $C_i = 10$, $C_p = 202$ and $C_f = 550$. The cost of producing nonconforming items is $C_{nc} = 100$. Failure threshold level of the component is set as $X_f = 5.15$. The constant production and demand rates are $p = 2$ and $d = 1$ respectively. Inventory holding cost per item per unit time $C_h = 5$. The setup cost per lot is $C_s = 50$. Time length of the preventive maintenance and corrective maintenance actions is $l = 1.39$.

A solution, $X_p^* = 2.49, t_0^* = 2.7263,\ \tau^* = t_0^*$, is obtained by the Frank-Wolfe algorithm. The model is solved for each inspection interval which are $\tau = t_0, 2t_0, \dots Mt_0$. Optimal values of X_p and t_0 are found for each inspection interval ($\tau = nt_0$) and the one that minimizes the long-run average cost rate is chosen as an optimal solution. The long-run average cost rate with respect to X_p and t_0 is shown in Figure 5.

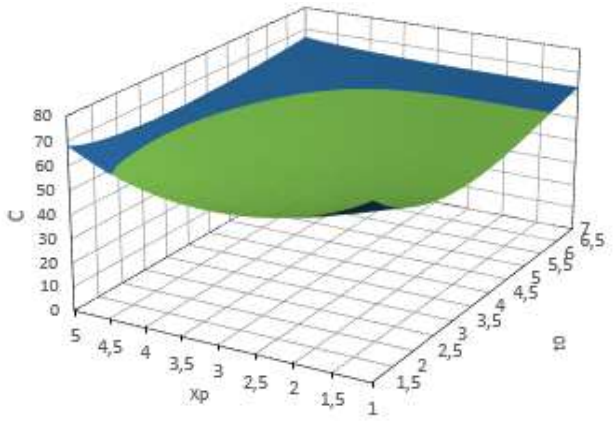


Figure 5. $C(\tau = t_0, t_0, X_p)$ with $X_f = 5.15,\ a = 1.2,\ b = 0.8$

CONCLUSION

In this study, a model is constructed for the joint optimization of lot-sizing and condition-based maintenance. Degradation process of the production system is modeled as a stationary Gamma process. Inspections are done periodically to observe the degradation level. Inspection cost is considerable so appropriate length of inspection period needs to be selected to minimize the overall cost rate. Maintenance actions are conducted in idle time periods in order not to interrupt the production plan. Renewal Reward Theory is used to compute the average long-run total cost rate. For a given τ, optimal values of X_p and t_0 are found by solving optimization problems with a nonlinear objective function and linear constraints. The Frank-Wolfe algorithm is used to solve this problem. Enumeration is done on τ to find the minimum value of the cost rate.

REFERENCES

Alaswad S., Xiang Y. (2017). A review on condition-based maintenance optimization models for stochastically deterioration system. Reliability Engineering and System Safety, 157, 54-63.

Ben-Daya M., Makhdoum M. (1998). Integrated production and quality model under various preventive maintenance policies. Journal of the Operational Research Society, 49, 840-853.

Ben-Daya M. (2002). The economic production lot-sizing problem with imperfect production process and imperfect maintenance. International Journal of Production Economics, 76, 257, 264.

Chen Y (2013). An optimal production and inspection modeling with minimal repair and rework considerations, Applied Mathematical Modeling, 37, 1618-1626.

Cheng Q. C., Zhou B. H., Li L. Joint optimization of lot sizing and condition-based maintenance for multi-component production systems, Computers & Industrial Engineering, 110, 538-549.

Hillier F. S. , Lieberman G. J. (2001). Introduction to Operations Research (Seventh Edition). McGraw-Hill.

Jafari L., Makis M. (2015). Joint optimal lot sizing and preventive maintenance policy for a production facility subject to condition monitoring, International Journal of Production Economics, 169, 156-168.

Jardine A.K.S, Lin D., Banjevic D (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance, Mechanical Systems and Signal Processing, 20, 1483-1510.

Khatab A., Diallo C., Aghezzaf E.H. and Venkatadri U. (2017). Optimizing the integrated economic production quantity for a stochastically deterioration production system under condition-based maintenance, 7^{th} IESM Conference.

Martin K.F (1994). A review by discussion of condition monitoring and fault-diagnosis in machine-tools, International Journal of Machine Tools and Manufacture 34 (1994) 527–551.

Peng, H. and Houtum, V. G. J. (2016). Joint optimization of condition-based maintenance and production lot-sizing. European Journal of Operational Research, 253, 94–107.

Van Noortwijk J. (2009). A survey of the application of gamma processes in maintenance. Reliability Engineering & System Safety, 94 , 2–21 .

BIOGRAPHIES

ALP DARENDELİLER is a doctoral student in the Industrial Engineering and Operations Research Program at the Department of Industrial Systems Engineering and Product Design, Faculty of Engineering and Architecture, Ghent University. He obtained his master degree in Industrial Engineering from Middle East Technical University, Turkey and bachelor degree in Industrial Engineering from Bilkent University, Turkey.

DIETER CLAEYS obtained the masters degree in Computer Science and the Ph.D. degree in Engineering in 2006 and 2011 respectively, both from Ghent University, Belgium. In 2007, he received the ORBEL award, which is accredited annually to the best Belgian master thesis in the field of Operations Research. From 2006-2017, he has been working at the SMACS Research Group within the Department of Telecommunications and Information Processing at Ghent University. From 2014-2017 he has been a postdoctoral research fellow from FWO. Since October 2015, he is an assistant professor at the department of Industrial Systems Engineering and Product Design at Ghent University. His main research interests include maintenance engineering, and the analysis of discrete-time queueing models and its applications to operations research and telecommunications systems.

ABDELHAKIM KHATAB is an Associate Professor at the University of Lorraine (France). He received a Ph.D. degree in Industrial Automation from the National Institute of Applied Science (INSA) of Lyon (France), in 2000. He held a researcher-lecturer position at INSA of Lyon from 2000-2004. After that (2005-2008), he held a post-doc position in Mechanical Engineering Department at Laval University, Quebec (Canada). His current research involves reliability design and analysis, and maintenance optimization of manufacturing and production systems.

EL-HOUSSAINE AGHEZZAF is full professor of industrial systems engineering and operations research at the Faculty of Engineering and Architecture of Ghent University. He holds a MS. and a Ph.D. in applied mathematics and operation research from CORE - Center for Operations Research and Econometrics at UCL-Université catholique de Louvain. He is currently heading the department of Industrial Systems Engineering and Product Design (ISyE-PD). His main research interests are in integrated systems optimization and simulation, stochastics and data analytics, supply and utility networks engineering.

ASSEMBLY LINE OPTIMIZATION

Optimization of a Supply Chain Game Orders with Order Cost

Florian Boyrie
Nicolas Pech-Gourg
Université Perpignan Via Domitia
52 Avenue Paul Alduy
66100 Perpignan, FRANCE
florian.boyrie@etudiant.univ-perp.fr
nicolas.pech-gourg@univ-perp.fr

Thibaud Serieye
Sciado Partenaires
31 Place Jules-Grandclément
69100 Villeurbanne, FRANCE
tserieye@sciado.fr

KEYWORDS

Supply Chain, Beer Game, Optimization, Simplex, Branch and Bound, Branch and Cut.

ABSTRACT

Supply chain management in the collaborative supply chain requires that all actors of the supply chain must understand the interests of all actors so that they get the best from the entire supply chain. The bullwhip effect is often a consequence of a misunderstanding among actors, and beer game shows exactly this effect. However, this game does not consider the order cost, meaning the cost a preparing an order. This order cost should disrupt the supply chain management, because each actor seeks to minimize this cost by reducing his number of orders. Therefore, suppliers get orders that are not linked to the demand of the end customer. We developed a game, based on beer game that includes this order cost, and we found a model to solve this optimization problem of the supply chain. In this paper, we present the simulator we used, our problem modeling, as well as the solver we created to solve the problem and we comment our results.

1 INTRODUCTION

The supply chain management is the subject of many studies and to understand it, a game has even been invented for MIT students by Jay Forester [5]. His game allows to illustrate the bullwhip effect [2]. We observe that when the retailer in the supply chain, increases a bit his order quantity, an amplification of the order quantity runs until the manufacturer.
In the Jay Forrester model, the beer game, there is no order cost, then we developed a game based on the beer game with this order cost. Thus, users are encouraged to order more than once instead of ordering every day. This choice has several consequences, they have to thing about their order cost, their storage cost increased by a higher stock, their supplier receives demands that are sometimes more important, which can disrupt their perception of the demand and force him to make a bullwhip effect thus ordering a bit more to his supplier.This order cost will therefore add more difficulties to the supply chain management. The decision is harder to take with the order cost but, if they want to minimize their costs, they must take it into account.
Thus, once the game is over, they will be awaiting feedback and explanations from the trainer about what they should have done. Our work will allow trainers to bring these answers to the decision makers. We made a model and resolved this problem to find the solution which (knowing the demand of the end customer) minimizes the total cost of the supply chain. In this article, we will present in a first part, the game that we developed with its particularity related to the order cost. We will then present the model of this problem and the steps to its resolution. In the last part, we will explain the results obtained on this problem, as well as our future work objectives further on.

2 THE DEVELOPED GAME

2.1 Setup

In the beer game, four players must take respectively the role of a retailer, a wholesaler, a distributor and a manufacturer of a beer supply chain. Depending on the customer consumption in the retailer, his stock and his orders in progress, the user who plays the retailer role has to think about his order for his supplier (distributor in the beer game), and so on to the manufacturer. Orders are delivered with a delay. Players must avoid the stockout on the supply chain and the overstock too. Of course, in both cases, a cost is linked with each situation.

2.2 Game Operation

The game we have developed consists in delivering soaps, on the supply chain, from the industrial to the end customer, as in the beer game. But we added an order cost linked with each order.

2.2.1 The Environment

Each player takes the place of a supply chain actor and must take his decision to deliver the product. His decision can be based on: the state of his stock, his stock at the beginning of the day, the demand of his customer, the orders he has already made, the different costs (order, storage), the history of the received orders. The player will have to honor all costumer's demands. Even if the player does not have the sufficient stock for the demand, he must deliver it as soon as soon as he will have the sufficient stock.

2.2.2 The Decisions

Each day, each supply chain actor must decide the quantity of his daily order to his supplier. To do so, he must understand his flow, study the demands of his customer and his stock so that he will not be stockout or overstock and thus avoiding the costs associated with these situations.
Once each actor takes his decision, the game calculates the new situation of each at the beginning of the following day. Each player must therefore decide again. The game stops when the trainer decides to take stock of the situation, he can either give an explanation and ask players to resume the game or stop them to close the game.

2.3 The Objective

Players will take a decision per day over a specified period. Their objective is to minimize the costs impacted by their decision. They will have to minimize the storage costs, the order costs and the stockout costs. The total of these costs is visible for all players throughout the trainer feedbacks, they can therefore compare costs of each team and discover which one manages the best their supply chain.

2.3.1 The Storage Cost

This is the cost of owning the stock. To calculate it, we use the total cost of ownership (TCO), which includes the cost of the product from acquisition to abandonment of this product.
The storage cost is proportional to the quantity of the available stock, to minimize this cost, we have to avoid overstock.

2.3.2 The Order Cost

This is the cost associated with each order, it allows to consider the costs related to the order management, purchase order, management of invoices, payment, delivery, quality controls... It is considered fixed for each order, regardless of the quantity ordered. This cost is taken into account only if the order quantity is more than zero.

2.3.3 The Outstock Cost

This is a penalty in case the player has not delivered his client demand. It is the cost applied in case of late delivery and it is proportional to the quantity that could not be delivered.

2.3.4 The Total Cost

The total cost is the clue that allows to compare the performance of each team, since it represents the sum of all the costs for the four players of each team.

3 RESOLUTION

Our problem is a linear programming problem with integer and binary variables. For the resolution we used the simplex algorithm with the Branch and Cut algorithm.

3.1 Algorithms Used

We solved the problem with the open source library GLPK[6], which proposes the simplex algorithm and the Branch and Cut method. The Branch and Cut method is based on the Branch and Bound method, cutting branches in the search tree that could not provide the optimal solution. The GLPK library offers various algorithms and methods associated with Branch and Cut. It is possible to choose among 4 cutting algorithms, 5 methods of branching new nodes in the tree and 4 methods of backtracking.

3.1.1 Cutting Algorithms And Branch And Cut

There are four different algorithms in the GPLK library: Gomory's mixed integer (GMI)[3] ; Mixed integer rounding (MIR)[2] ; Mixed cover cut (COV)[6] ; Clique cut (CLQ)[6]
We compared these algorithms (B&B, GMI, MIR, COV, CLQ) by testing them over several days (between 10 and 18 days). For each test, we measured the resolution time in seconds. The results are shown in the table below:

Table 1: Algorithms comparison (seconds)

Days	B&B	MIR	COV	CLQ	GMI
13	0,476	0,579	0,484	0,528	**0,384**
14	**1,118**	1,347	1,162	1,199	1,487
15	2,788	3,544	3,029	3,091	**2,606**
16	NI	NI	NI	NI	**22,61**
17	NI	197,2	NI	NI	**3,941**
18	46,88	31,96	42,16	40,39	**3,878**

*NI : superior to 600 seconds

During these tests the other two parameters, of which we will talk about later, were at the default value (DTH for the branching and BLB for the backtracking).
For our problem, we can see that the algorithm Gomory's mixed integer is the faster.

3.1.2 Branching Method

There are five branching methods in the GLPK library: First fractional variable (FFV) ; Last variable (LFV) ; Most variable (MFV) ; Heuristic by Driebeck and Tomlin (DTH)[1] ; Hybrid pseudo-cost heuristic (PCH)[1]

Table 2: Branching methods comparison (seconds)

Days	FFC	LFV	MFV	DTH	PCH
13	106,4	8,196	4,819	0,384	**0,111**
14	58,66	5,652	46,78	1,487	**0,17**
15	NI	341,9	185,2	2,606	**0,753**
16	NI	NI	NI	22,61	**3,875**
17	NI	NI	NI	3,941	**0,801**
18	NI	NI	NI	3,878	**1,222**

*NI : superior to 600 seconds

During these tests, the other two parameters were at GMI for the algorithm that cuts the branches and at the default value BLB for the backtracking. We can see that the

Hybrid pseudo-cost heuristic method is the faster on our problem.

3.1.3 Backtracking Technique

There are four backtracking techniques in the GLPK library:

- Depth first search (DFS)
- Breath first search (BFS)
- Best local bound (BLB)[4]
- Best projection heuristic (BPH)[4]

Table 3: Backtracking techniques comparison (seconds)

Days	DFS	BFS	BLB	BPH
13	0,336	0,158	**0,078**	0,085
14	0,25	0,223	0,121	**0,108**
15	0,488	1,187	0,535	**0,359**
16	2,351	4,129	2,729	**2,294**
17	0,792	0,733	0,564	**0,391**
18	1,035	1,791	0,856	**0,532**
19	**36,28**	NI	77,79	65,15
20	40,01	34,44	52,84	**21,45**

*NI : superior to 600 seconds

In these tests the other two parameters were at the value (GMI for the cutting algorithm and PCH for the branching). We can see that the Best projection heuristic method gets faster results for our problem.

3.2 Results

We got our fastest results with the settings below:

- Cutting algorithm: Gomory's mixed integer
- Branching method: Hybrid pseudo-cost heuristic
- Backtracking technique : Best projection heuristic
- Feasibility pump heuristic: ON (improving the speed)
- MIP pre-solver: ON (improving the speed)

The fastest result obtained is 33 days of simulation in about 3 hours of execution for one scenario game. However, this result is nuanced because we are working on a dataset. We can see on this graph the effect of the Branch and Cut with an evolution which is not constant according to the number of branches which are cut and to the different parameters dimensioned previously. From day 27, the execution time becomes very irregular and dependent on the dataset used.

4 CONCLUSIONS AND PERSPECTIVES

In this paper, we presented a game, based on the beer game, which allows to understand the difficulties related to supply chain management and mainly to the bullwhip effect. We added the order cost to this game, which made the management harder, and allowed the game to be a bit more realistic. We then found the problem and sought

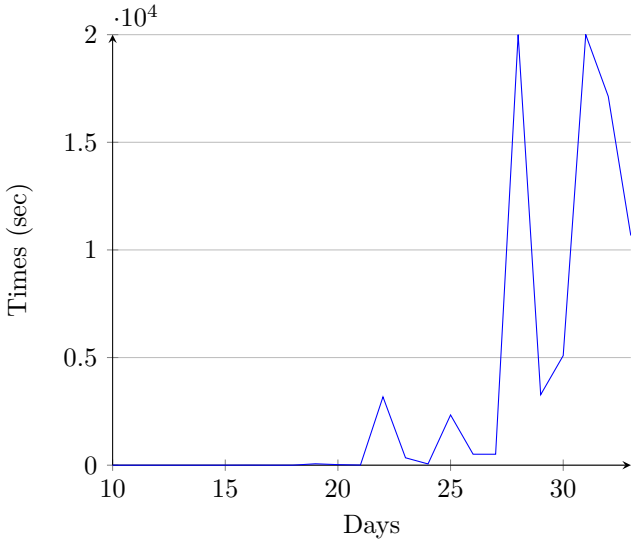


Figure 1: Best algorithm performances

methods that could solve it over the highest number of days possible, while maintaining the guarantee that the solution found was optimal. Indeed, this game is used by trainers who want, once the game is over, to show the players what would have been the best solution. We used the Branch and Cut approach, and at first we compared several cutting algorithms. We then compared several methods of branching and backtracking. The fastest solution found is to answer our problem for 33 days in 10 669 s. The goal is achieved since the trainers who animate the game can now present the optimal solution to the players and explain what should have been done. Players can also compare their results to the optimal solution. The next step will be to parallelize our solution to get a result faster.

REFERENCES

[1] T. Achterberg, T. Koch, and A. Martin. Branching on history information. Technical Report 02-32, ZIB, Takustr. 7, 14195 Berlin, 2002.

[2] A. Atamtürk and V. Narayanan. Conic mixed-integer rounding cuts. *Mathematical Programming*, 122(1):1–20, Mar 2010.

[3] A. Caprara and M. Fischetti. {0, 1/2}-chvátal-gomory cuts. *Mathematical Programming*, 74(3):221–235, Sep 1996.

[4] E. L. J. E.Mitchell. Branch-and-bound method for integer programming. 1998.

[5] J. W. Forrester. Industrial dynamics. *Journal of the Operational Research Society*, 48(10):1037–1041, Oct 1997.

[6] A. Makhorin. Glpk (gnu linear programming kit).

THE INFLUENCE OF LINE BALANCING ON LINE FEEDING FOR MIXED-MODEL ASSEMBLY LINES

Hendrik Wijnant
Nico André Schmid
Veronique Limère
Department of Business Informatics and Operations Management
Ghent University, Belgium
Flanders Make, Lommel, 3920, Belgium
E-mail: nico.schmid@ugent.be

KEYWORDS
Decision Making, Optimization, System Management

ABSTRACT

Though, recent research on mixed-model Assembly Line Balancing Problems (MALBP) and Assembly Line Feeding Problems (ALFP) aims to incorporate real-world aspects, research on the integration of both areas is still limited. This paper helps closing this gap by studying the influence of different balancing objectives on line feeding decisions and costs. For line balancing, different objective functions were formulated and the results were used as input when solving the ALFP. Although, no large cost differences were found, we observed that decision making in line feeding does depend on the balance.

INTRODUCTION

Nowadays, a rising number of models is produced on mixed-model assembly lines (Schmid et al. 2018) while each product requires its specific parts and the organization of tasks differs. Consequently, a large variety of parts has to be supplied to the assembly line, resulting in high storage space requirements at the Border of Line (BoL) being the area where parts are stored before usage. This leads to an increasing relevance of the material supply configuration for the effectiveness of the production system.

In the Assembly Line Feeding Problem (ALFP), decisions on provision and storage of parts at the BoL are taken by assigning them to policies such as line stocking (providing parts on a pallet) or kitting (presorting multiple parts in a smaller container). Mostly, the ALFP is solved after tasks have been assigned to individual assembly stations by solving the Mixed-model Assembly Line Balancing Problem (MALBP) (Sternatz 2015). As the latter determines the amount of tasks and, therefore, parts at every station it affects decision making in line feeding. In a MALBP, several objectives may be used to optimize a line, thereby resulting in different line configurations. Most research deals with both problems separately and also in practice those two optimization problems are solved successively. But, as there are attempts to integrate decision making in line balancing and feeding (Battini et al. 2016, Sternatz 2015), it is of strong interest to investigate the effect of varying objective functions in line balancing on the selection of line feeding policies and the corresponding costs as this has not been done so far.

Our results indicate that, even balancing lines in a pretty different way, effects on line feeding costs are marginal. However, it is found that the actual assignment of parts to line feeding policies is varying with the chosen balancing objective function to a large extent. Furthermore, some line feeding policies are more robust against changing objective functions than others.

The remainder of this research is organized as follows. First, a literature review is provided. Next, MIP models for line balancing and feeding are formulated. Thereafter, preliminary results are described and discussed. In the last section, a conclusion is drawn.

LITERATURE REVIEW

Mixed-model assembly line balancing problems

In MALBP, tasks are distributed among working stations with respect to some cost and/or capacity objective. The best known problem formulations are MALBP-F, MALBP-1, MALBP-2, and MALBP-E (Becker and Scholl 2006). In the MALBP-F, both, cycle time and number of stations are given and a feasible line balance has to be found. In MALBP-1, cycle time is given and the number of stations is minimized whereas in the MALBP-2, the number of stations is given and the cycle time is minimized. In MALBP-E, a line's efficiency is maximized by minimizing both, cycle time and number of stations. Finally, for smoothing the workload, one can smooth varying times for different models at ev-

ery station, i.e. horizontal balancing (Becker and Scholl 2006). For a classification of research on the ALBP the reader is referred to (Boysen et al. 2007).

Assembly line feeding problem

According to (Kilic and Durmusoglu 2015) part feeding systems have three main components: parts storage, parts transportation and feeding policy. However, preparation of parts might also be taken into account. In literature, five different line feeding policies are distinguished (Schmid et al. 2018). In line stocking, homogeneously packed parts, e.g. on a pallet, are directly transported from a warehouse to the BoL. In kanban, parts are repacked into smaller, homogeneously filled, bins. Sequencing describes that different variants of a part presorted in containers according to the sequence of consumption. Kitting, as an extension of sequencing combines various components and their variants in a container. Stationary kits only contain parts required at a single station, whereas traveling kits contain parts for multiple stations and travel together with the product along the assembly line.

(Limere et al. 2012) developed a decision model to determine the optimal line feeding considering two feeding policies, line stocking and stationary kitting. (Caputo et al. 2015) developed an IP model for finding an optimal feeding policy mix (line stocking, kitting or JIT, i.e. kanban) in a single-model assembly line. (Schmid et al. 2018) developed a MIP cost minimization model to determine the optimal feeding policy for every individual part in a mixed-model assembly line. This model considers the, above described, five policies. Traveling kits are placed on an assembly line's start and removed at the end, thus allowing only one kit per product.

Integration of line balancing and line feeding

(Sternatz 2015) finds potential productivity gains by simultaneously solving line balancing and feeding problems considering direct (line stocking) and indirect supply (stationary kitting). He states that indirect supply can be avoided and hence logistical costs are reduced when decisions are integrated. (Battini et al. 2016) formulate a similar model, additionally including ergonomic aspects. They also distinguish direct part feeding and indirect part feeding and confirm productivity gains as described by (Sternatz 2015). Both models minimize the number of required assembly and supermarket operators, having a fixed cycle time. Therefore, it seems promising, to investigate different objective functions' effect on line feeding.

MATHEMATICAL MODELS

We formulated four MIP balancing model with different objective functions and used the results to apply the ALFP model described by (Schmid et al. 2018) as well as an extension, allowing multiple traveling kits.

Mixed-model assembly line balancing problem

In this section, four balancing models are formulated, i.e. MALBP-1, MALBP-2, MALBP-E and MALBP-E with horizontal balancing. Each MALBP is reduced to a simple form by making the following assumptions.

1. Serial (straight) assembly line layout.
2. Paced assembly line.
3. Deterministic task times.
4. Only precedence constraints.
5. Demand for all products is equal.

Following notations are used.

S	Set of stations, index s
P	Set of products, index p
J	Set of tasks, index j
t_j	Task time for task j
$Pred_j$	Set of direct predecessors of task j
λ_{jp}	1 if task j is needed for product p, 0 otherwise
c	Cycle time
m	Total number of stations installed
DS	Set of definite stations, $DS = \{1,...,\underline{m}\}$
PS	Set of probable stations, $PS = \{\underline{m}+1,...,\overline{m}\}$

MALBP-1
The number of stations is minimized (equation (1)) while the cycle time is constant. Two binary decision variables are used: y_s equals 1 if station s is installed and x_{js} equals 1 if task j is assigned to station s.

$$\min \sum_{s \in S} y_s \tag{1}$$

$$\text{s.t.} \sum_{j \in J} t_j \cdot x_{js} \cdot \lambda_{jp} \leq c \cdot y_s \qquad \forall s \in S, p \in P \tag{2}$$

$$\sum_{s \in S} x_{js} = 1 \qquad \forall j \in J \tag{3}$$

$$\frac{\sum_{\substack{w \in S \\ w \leq s}} \sum_{h \in Pred_j} x_{hw}}{|Pred_j|} \geq x_{js} \qquad \forall s \in S, \forall j \in J \tag{4}$$

$$y_{s-1} \geq y_s \qquad \forall s \in S \tag{5}$$

Constraint (2) ensures that the station load for any product does not exceed the cycle time, as assumption 2 specifies a paced assembly line. Furthermore, constraint (2) ensures that tasks can only be assigned to installed stations. Constraint (3) guarantees that all tasks are assigned to exactly one station while constraint (4) enforces precedence relations. Finally, constraint (5) was added for adjacency of all installed stations.

MALBP-2

Opposite to MALBP-1, the cycle time is minimized (equation (6)) while the number of stations is constant. This approach maximizes productivity.

$$\min\ c \tag{6}$$

$$\text{s.t.} \sum_{j\in J} t_i \cdot x_{js} \cdot \lambda_{jp} \leq c \qquad \forall s \in S, p \in P \tag{7}$$

$$(3),\ (4)$$

Constraint (7) is similar to constraint (2) and makes sure that station loads do not exceed the cycle time. The remaining constraints are the same as before.

MALBP-E

The proposed model for the MALBP-E below is an extension of the formulation given by (Esmaeilbeigi et al. 2015). In SALBP-E, line efficiency is maximized by minimizing both cycle time and number of stations. Similarly, in MALBP-E, the weighted average line efficiency, based on the product demand, is maximized. (Esmaeilbeigi et al. 2015) prove that the line efficiency is maximized by minimizing line capacity $T = c \cdot m$, alternatively defined as $T = t_{sum} + \delta_{total}$, with t_{sum} denoting the sum of all task times and δ_{total} denoting the total idle time over all stations. As we assume equal demand for all products (assumption 5), idle times of a station for a product, and the average station idle time, denoted as δ_{sp} and δ_s respectively, can be calculated as

$$\delta_{sp} = c - \sum_{j\in J} t_j \cdot \lambda_{jp} \qquad \forall p \in P,\ \forall s \in S \tag{8}$$

$$\delta_s = \frac{\sum_{p\in P} \delta_{sp}}{|P|} \qquad \forall s \in S \tag{9}$$

The line efficiency can thus be calculated as

$$T \cdot |P| = \sum_{p\in P} \big(\sum_{s\in S} \delta_{sp} + \sum_{j\in J} t_j \cdot \lambda_{jp} \big), \tag{10}$$

which is equivalent to

$$T = \sum_{s\in S} \delta_s + \frac{\sum_{p\in P} \sum_{j\in J} t_j \cdot \lambda_{jp}}{|P|} \tag{11}$$

Since t_j, λ_{jp} and $|P|$ are parameters, minimizing the line capacity T is equivalent to minimizing the sum of average idle times over all stations. Hence, a similar linearization method as proposed by (Esmaeilbeigi et al. 2015) can be used. Summarizing, the following decision variables are used: x_{js}, y_s, c and δ_{sp}.

$$\min \sum_{s\in S} \sum_{p\in P} \delta_{sp} \tag{12}$$

$$\text{s.t.} \sum_{j\in J} t_j \cdot \lambda_{jp} \cdot x_{js} + \delta_{sp} = c \quad \forall s \in DS,\ \forall p \in P \tag{13}$$

$$\sum_{j\in J} t_j \cdot \lambda_{jp} \cdot x_{js} + \delta_{sp} \leq c \quad \forall s \in PS,\ \forall p \in P \tag{14}$$

$$\sum_{j\in J} t_j \cdot \lambda_{jp} \cdot x_{js} + \delta_{sp} \geq c + \overline{c} \cdot (y_s - 1) \qquad \forall s \in PS,\ \forall p \in P \tag{15}$$

$$\sum_{j\in J} t_j \cdot \lambda_{jp} \cdot x_{js} + \delta_{sp} \leq \overline{c} \cdot y_s \qquad \forall s \in PS,\ \forall p \in P \tag{16}$$

$$y_s = 1 \qquad \forall s \in DS \tag{17}$$

$$x_{is} \leq y_s \qquad \forall j \in J,\ \forall s \in S \tag{18}$$

$$\underline{c} \leq c \leq \overline{c} \tag{19}$$

$$(3),\ (4),\ (5)$$

The objective function (12) seeks to minimize the total idle time over all products and stations. Constraint (13), (14), (15) and (16) make sure that the station load for a product does not exceed the cycle time (assumption 2). For these constraints, a lower and upper bound on both the cycle time, $\underline{c}$ and $\overline{c}$, and the number of stations, $\underline{m}$ and $\overline{m}$ are specified. Next, constraint (17) stipulates that all stations in the set of definite workstations should be installed and constraint (18) makes sure that tasks can only be assigned to installed stations. Finally, constraint (19) restricts the value for the cycle time to lie between its bounds.

MALBP-E with horizontal balancing

In order to balance the varying workload of the stations, caused by different models, we added horizontal balancing to the MALBP-E. (Thomopoulos 1970) proposed an objective function that minimizes the sum of the absolute deviation of the station time of a product from the average station time, i.e. $\sum_{s\in S} \sum_{p\in P} |t_{ps} - t_s|$. This objective function value was linearized using variables v_{sp} and w_{sp}, and added to the objective function of the MALBP-E. The decision variables are: x_{js}, y_s, c, δ_{sp}, v_{sp} and w_{sp}.

$$\min \sum_{s\in S} \sum_{p\in P} (\delta_{sp} + l \cdot (v_{sp} + w_{sp})) \tag{20}$$

$$\text{s.t. } v_{sp} \geq 0,\ w_{sp} \geq 0 \qquad \forall p \in P, \forall s \in S \tag{21}$$

$$v_{sp} \geq t_{ps} - t_s \qquad \forall p \in P, \forall s \in S \tag{22}$$

$$w_{sp} \geq t_s - t_{ps} \qquad \forall p \in P, \forall s \in S \tag{23}$$

$$(3),\ (4),\ (5),\ (13),\ (14),\ (15),\ (16),\ (17),\ (18),\ (19)$$

$$\text{with } t_{ps} = \sum_{j\in J} \lambda_{jp} \cdot x_{js} \cdot t_j, \qquad \forall p \in P, \forall s \in S \tag{24}$$

$$t_s = \frac{\sum_{p\in P} t_{ps}}{|P|} \qquad \forall s \in S \tag{25}$$

In objective function (20), the total absolute deviation from the average station time over all models and stations is added to the MALBP-E objective function. A

weight l is added to enable specifying the importance of horizontal balancing compared to the line efficiency maximization. Constraints (21), (22) and (23) linearize the absolute deviation.

Assembly line feeding problem

In this section, two models for solving the ALFP are discussed. Firstly, the model formulated by (Schmid et al. 2018) is used. It optimizes the line feeding configuration of a mixed-model assembly line while incorporating space adjustments of stations and considers all steps from storage to final assembly. For a more detailed description of the model, the reader is referred to (Schmid et al. 2018).

To allow multiple traveling kits, the model of (Schmid et al. 2018) was extended by including the possibility of taking out a depleted traveling kit and inserting a full one at any station of the line. An extra notation is introduced.

ft^T Number traveling kit batches needed at the BoL

Furthermore, extra binary decision variables were introduced.

y_f^T 1 if family f is fed in a traveling kit, else 0
y_{pfs}^T 1 if family f travels in a traveling kit along with product p and is removed at station s, else 0
y_{ps}^T 1 if a traveling kit is used for product p that needs to be retrieved at station s, else 0
y_s^T 1 if a traveling kit is replaced at station s, else 0
y_p^T 1 if at least one traveling kit is used for product p, else 0

The MIP formulation was extended by adding the following cost elements: transportation costs for every traveling kit, and usage costs for replacing an empty traveling kit with a full one.

$$CT^T = \frac{ft^T \cdot di^T + \left(nt^T - ft^T\right) \cdot mr^T}{nbc^T \cdot \mu^T \cdot VV^T} \tag{26}$$

$$ft^T = \frac{\sum_{p \in P} y_p^T \cdot d_p}{bs^T} \tag{27}$$

$$nt^T = \frac{\sum_{p \in P} \sum_{s \in S} y_{ps}^T \cdot d_p}{bs^T} \tag{28}$$

For the transportation cost in equation (26), it is assumed that traveling kits inserted at the beginning of the line (calculated in equation (27)) are transported by forklifts, whereas the replacement kits (calculated in equation (28)) are transported to the stations in milk runs.

$$CU^T = \sum_{p \in P} \sum_{s \in S} d_p \cdot y_{ps}^T \cdot \left(2 \cdot ht^T + \frac{2 \cdot de}{OV}\right) \tag{29}$$

The usage cost calculation in equation (29) is similar to the calculation of the usage cost for stationary kits, however the handling time ht^T is counted double as both the empty and new kit have to be handled. As traveling kits are replaced at the end of the station (see equation(34)), a walking distance only equaling two times the distance de between station and BoL needs to be covered.

Furthermore, some additional constraints were added.

$$\begin{aligned} &D_s \cdot l^D + \sum_{i \in I} x_{is}^L \cdot l^L + \sum_{f \in F} x_{fs}^S \cdot l^S \\ &+ x_s^K \cdot l^K + y_s^T \cdot l^T \leq EP_s - SP_s \qquad \forall s \in S \end{aligned} \tag{30}$$

Firstly, by adapting constraint (30), storage space at the BoL for the replacement traveling kits is reserved.

$$\sum_{f \in BOM_p} y_{pfs}^T \cdot v_f \leq V^T \cdot y_{ps}^T \quad \forall p \in P, \forall s \in S \tag{31}$$

Secondly, equation (31) enforces that the volume of the parts in the traveling kit does not exceed the kit volume.

$$y_{pfu}^T = 0 \qquad \forall s \in S, \forall f \in F_s, \forall u \in S : u < s \tag{32}$$

Next, by adding constraint (32), we assure that a traveling kit is not taken out of the line when it still holds parts needed at stations downstream.

$$\begin{aligned} y_f^T + y_{pgu}^T - 1 \leq y_{pfu}^T \quad &\forall p \in P, \forall s \in S, \forall f \in BOM_p \cap F_s, \\ &\forall q \in S : q \leq s, \forall g \in BOM_p \cap F_q, \\ &\forall u \in S : u \geq s \end{aligned} \tag{33}$$

Constraint (33) ensures that only one traveling kit can travel along with the product at the same time.

$$y_f^T \cdot + y_{ps}^T - 1 \leq y_{pfs}^T \quad \forall p \in P, \forall s \in S, \ \forall f \in BOM_p \cap F_s \tag{34}$$

Constraint (34) enforces that a newly inserted travelling kit does not contain any parts needed at the station of insertion. By adding this limitation, a new travelling kit is forced to be inserted near the end of the station.

Lastly, some additional auxiliary constraints were added.

$$y_f^T \cdot |V_f| \geq \sum_{i \in V_f} \sum_{s \in S} x_{is}^T \qquad \forall f \in F \tag{35}$$

$$\sum_{s \in S} y_{pfs}^T - y_f^T = 0 \qquad \forall p \in P, \forall f \in BOM_p \tag{36}$$

$$\begin{aligned} &y_{ps}^T \cdot |BOM_p| \\ &\geq \sum_{f \in BOM_p} y_{pfs}^T \qquad \forall p \in P, \forall s \in S \end{aligned} \tag{37}$$

$$y_p^T \cdot |S| \geq \sum_{s \in S} y_{ps}^T \qquad \forall p \in P \quad (38)$$

$$y_s^T \cdot |P| \geq \sum_{p \in P} y_{ps}^T \qquad \forall s \in S \quad (39)$$

The model specified by (Schmid et al. 2018) is referred to as ALFP with single traveling kit and it's extension as ALFP with multiple kits.

PRELIMINARY RESULTS

First, all balancing models are optimized to obtain a line configuration. Secondly, both line feeding models were solved for every obtained line balance. All mathematical models are implemented and solved using CPLEX (A time limit of 3600s yielded to average LP-gaps of 1.5% and maximal gaps of 15.1%).

For testing purposes, 2 sources of data were merged in order to create 16 instances. For balancing, datasets from (Scholl 1993) and instances generated by the NTI-GEN software (Serrano et al. 2014), were used. For the ALFP in a mixed-model environment, 16 instances from (Schmid et al. 2018), were used. Both data sources were merged by linking tasks with part families (describing all variants of a part). Since the balancing related instances from literature only provide precedence links for a single product assembly line, we merged precedence constraints of two products.

Assembly line balancing

The MALBP-E model with horizontal balancing was solved with different weights l, i.e. 100% and 50% (see equation (20)). The former is denoted as 'MALBP-E+1HB' and the latter as 'MALBP-E+0.5HB'.

As can be seen in table (1), MALBP-2 balances logically comprise more stations, and consequently less tasks per station, than the other balances. Admittedly, it must be said that the resulting balance of MALBP-E depends on the chosen bounds for cycle time and number of stations.

Table 1: Line characteristics for different ALB objective functions (average)

	MALBP				
	1	2	E	E+1HB	E+0.5HB
Stations	13.4	19.7	13.9	12.9	13.1
Tasks/station	37.2	21	32.7	35	33.3

Assembly line feeding with single traveling kitting

Table (2) reveals that the maximum difference in costs only amounts to a maximum of 6% on average comparing MALBP-2 and MALBP-E+0.5HB.

Table 2: Line feeding costs (average over 16 instances)

	MALBP				
	1	2	E	E+1HB	E+0.5HB
Cost/part	97.82	95.74	95.93	98.95	101.25
st.dev.	31.98	32.58	28.30	31.10	32.75

It can be seen that a similar line feeding policy mix is used for lines resulting from MALBP-1, MALBP-E and MALBP-E with horizontal balancing (see table (3)). Whereas in MALBP-2 more stations, and hence more storing space is available,allowing more space consuming policies such as line stocking and sequencing.

Table 3: Average part feeding policy mix

	MALBP				
	1	2	E	E+1HB	E+0.5HB
Line stocking	19%	31%	10%	9%	31%
Kanban	26%	37%	15%	7%	15%
Sequencing	19%	26%	11%	16%	28%
St. kitting	17%	28%	10%	13%	32%
Tr. kitting	17%	30%	11%	13%	29%

Some parts are assigned to the same policy irrespective of the underlying balance (see table (4): L denotes line stocking, D kanban, S sequencing, K stationary kitting and T traveling kitting). Overall, 28.2% of all parts are always assigned to the same feeding policy. Furthermore, one can observe that parts, assigned to line stocking in all ALB solutions, have on average a considerable higher volume and demand than parts that are assigned to other policies, whereas it seems to be the opposite for sequenced parts. Parts in stationary kit switch policies in most cases (92.4%), indicating that it is the least favorable option.

Table 4: Characteristics of parts assigned to the same policy in all ALB solutions

	L	D	S	K	T
Same policy [%]	35.8	23	29.7	7.6	36.8
Avg part volume [dm^3]	37.6	5.7	3.3	3.1	7.3
Avg part demand [%]	32.5	0.5	0.7	45	7

Table (5) shows the change of a part's feeding policy from being supplied with one policy in a certain ALB solution to another policy in another ALB solution. For example, it can be seen that on average 14% of the parts that were line stocked in one ALB solution, are provided with kanban in another ALB solution.

Assembly line feeding with multiple traveling kitting

For all instances that could be solved close to optimality (12 out of 16), exactly the same results as when using

Table 5: Average changing behaviour of the part feeding policy assignments

		To				
		L	D	S	K	T
From	L	36%	14%	3%	7%	7%
	D	21%	59%	12%	28%	13%
	S	2%	4%	58%	12%	6%
	K	4%	11%	12%	36%	7%
	T	10%	11%	15%	17%	67%

a single traveling kit were found. However, a serious reduction of available space indicated some benefits when multiple kits were used.

Discussion

As we aimed to understand the effect of different balancing methods on line feeding, we found that balancing does influence decision making in line feeding, costs however, are not as strongly affected as expected (Sternatz 2015). This is probably due to the use of five line feeding policies providing flexibility in the assignments of parts to line feeding policies. Furthermore, space borrowing might also have affected this outcome. Therefore, not allowing space borrowing might show different results. As some models are not solved to optimality, numbers might change slightly, when solved to optimality.

So far, no benefits of using multiple traveling kits could be found in the conducted experiments. This is probably reasoned in the small datasets used as well as in large containers for traveling kits. Although it could not be proven, we expect that cost savings will outweigh fixed costs for traveling kits in larger instances and multiple traveling kits become an interesting option.

CONCLUSION

This research analyzed the influence of using different objective functions in the MALBP on the line feeding policy decisions, which varied for 72% of the parts, and the resulting line feeding costs, varying at most up to 6% . We also found that stationary kits seem to be the least favourable option and, therefore, the assignment of parts to stationary kits is less robust. As the scope of this research is limited to small assembly lines with two products, more research, in terms of computational experiments, is needed to gain a better insight in more advanced production lines. Additionally, revealing the influencing factors for decision making on line feeding policy assignment might be worthwhile. Furthermore, this research indicates that the possibility of using additional traveling kits is not beneficial in small assembly lines. However, more research is required to evaluate the usage of multiple traveling kits in larger production systems.

REFERENCES

Battini D.; Calzavara M.; Otto A.; and Sgarbossa F., 2016. *The Integrated Assembly Line Balancing and Parts Feeding Problem with Ergonomics Considerations. Ifac Papersonline*, 49, no. 12, 191–196.

Becker C. and Scholl A., 2006. *A survey on problems and methods in generalized assembly line balancing. European Journal of Operational Research*, 168, no. 3, 694–715.

Boysen N.; Fliedner M.; and Scholl A., 2007. *A classification of assembly line balancing problems. European Journal of Operational Research*, 183, no. 2, 674–693.

Caputo A.C.; Pelagagge P.M.; and Salini P., 2015. *A decision model for selecting parts feeding policies in assembly lines. Industrial Management and Data Systems*, 115, no. 6, 974–1003.

Esmaeilbeigi R.; Naderi B.; and Charkhgard P., 2015. *The type E simple assembly line balancing problem: A mixed integer linear programming formulation. Computers and Operations Research*, 64, 168 – 177.

Kilic H.S. and Durmusoglu M.B., 2015. *Advances in assembly line parts feeding policies: a literature review. Assembly Automation*, 35, no. 1, 57–68. ISSN 0144-5154. doi:10.1108/AA-05-2014-047.

Limere V.; Van Landeghem H.; Goetschalckx M.; Aghezzaf E.H.; and McGinnis L.F., 2012. *Optimising part feeding in the automotive assembly industry: deciding between kitting and line stocking. International Journal of Production Research*, 50, no. 15, 4046–4060.

Schmid N.; Limere V.; and Raa B., 2018. *Line Feeding with variable space constraints for mixed-model assembly lines. IFAC PapersOnLine*, 55, no. 11, 164–169.

Scholl A., 1993. *Data of Assembly Line balancing Problems. Schriften Zur Quantitativen Betriebswirtschaftslehre*, 16, 93.

Serrano M.C.; García Ó.C.; Arroyo S.D.; and Bautista J., 2014. *NTIGen: a software for generating Nissan based instances for time and space assembly line balancing.* In *Managing Complexity*, Springer. 121–128.

Sternatz J., 2015. *The joint line balancing and material supply problem. International Journal of Production Economics*, 159, 304–318.

Thomopoulos N.T., 1970. *Mixed model line balancing with smoothed station assignments. Management science*, 16, no. 9, 593–603.

Implementation of a Cyber-Physical Systems Simulation Components allocation tool

Henrick Deschamps and Gerlando Cappello
Airbus Operation SAS
Modelling and Simulation department
316 Route de Bayonne, 31300 Toulouse, France
Email: {firstname.name}@airbus.com

Janette Cardoso and Pierre Siron
ISAE-SUPAERO, University of Toulouse
10 avenue Édouard Belin, 31055 Toulouse, France
Email: {firstname.name}@isae-supaero.fr

KEYWORDS
Aeronautics, CPS, Modelling, Simulation, Allocation, Scheduling, HLA, CERTI.

ABSTRACT

This paper presents ongoing work on the formalism of Cyber-Physical Systems (CPS) simulations. In this paper, we consider distributed CPS simulations, for which there are strong constraints on the interaction of simulation components. In our previous work, we suggested a method to estimate a simulation scheduling on a given architecture and to verify constraints *a priori*. Trying to integrate these components manually is very time-consuming, and can lead to mistakes. We introduce a tool to make an automatic allocation and scheduling of CPS simulation. We base this on the real-time scheduling literature and adopt a heuristic-based approach, adapted to our framework. We then present a complete case study, including simulations and the physical architecture of the simulations, and we illustrate the generation of the scheduling with different heuristics.

Introduction

A Cyber-Physical System (CPS) is a feedback system comprised of communicating real-time systems and humans or environment in the loop.
Aircraft, Airbus and ISAE-SUPAERO activity area, is a type of CPS, where pilots, avionics, aircraft surfaces and the aircraft environment are tightly interacting through control loops to stabilize the vehicle.
CPS design work-flow can integrate simulation phases; this is particularly true in the aeronautical sector, different steps of the CPS design are illustrated in fig. 1. Development cycles are long and expensive on avionics product. They can be shortened using simulation during development and integration life-cycle.
Due to the complexity of the simulated systems and the simulated environment, as well as the need to incrementally improve systems, simulations are more and more modular. In our method, we consider that every modular component of the simulation is sufficiently representative. Furthermore, in our cases study, we only considered CPS with the physical part discretized, with

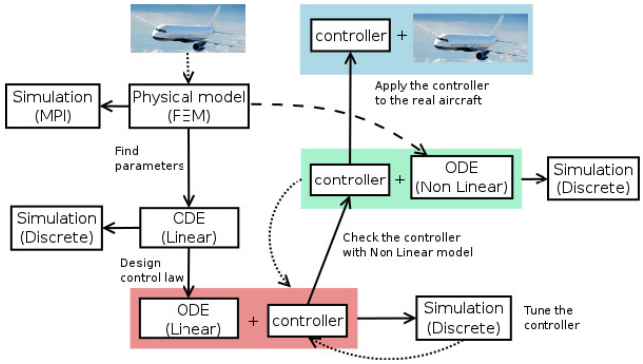


Figure 1: Steps of a CPS design and implementation

linear and non-linear ODEs, as depicted in fig. 2. For that, we rely on the existing skills of model engineering. These skills can be different if the model represents a physical part or a cyber part. Also, we do not address the problem of parallelization, or of the distribution of large model simulations. Our starting point is a set of components produced by experts in model engineering and distributed simulation engineering. With these components, specific abstracted constraints and degrees of freedom are expressed for the integration.

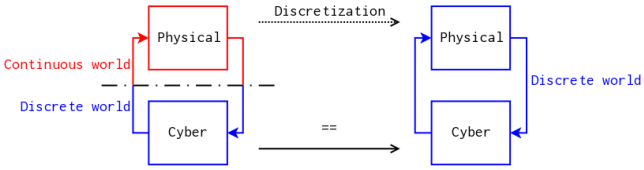


Figure 2: CPS discretization considered in our study

With these hypotheses, our problem is the composition of the existing simulation components that respect the abstracted constraints. In (Deschamps et al. 2017) and (Deschamps et al. 2018), a formalism for expressing the CPS simulation has been introduced. With small examples, temporal constraints can be verified on a CPS simulation, manually. Nevertheless, in an industrial context, a simulation of CPS can be composed of a substantial number of components and constraints. Our problem is the specification and implementation of tools that can automate the verification of constraints and the

generation of scheduling from the description of a CPS simulation.
In the following, after the related work, an overview of the CPS scheduling formalism is presented. Then we present our framework and will focus on the *Allocation function*, with an illustration using a concrete case-study. Finally, we will give concluding remarks and our future work.

Related work

Latterly, progress has been made in the study of accurate CPS simulation using modular blocks.
Numerous studies on CPS simulation in which the emphasis is on fidelity or verification uses the Functional Mock-up Interface (FMI) standard (Blockwitz et al. 2012). The FMI standard defines an implementation interface, the Functional Mock-up Unit (FMU), allowing the integration of heterogeneous models. This standard is therefore ideally suited for CPS simulation. An instance of FMU, also called a slave, can have a specific solver and be executed standalone or can wait for an execution directed by a master. The master algorithm also synchronizes FMUs, following specific synchronization points, and manages communication between FMUs.
In (Sadvandi et al. 2018), the authors present a CPS simulation platform capable of executing simulation loops with models, software, or hardware in the loop. These platform objectives are to meet the same needs as those presented in the introduction, namely the reduction of development costs through early validation through simulation. In this study, the constraints inherent to the CPS are highlighted, particularly at the level of the temporal behavior of the simulation integration, to validate a whole control system. FMI was chosen to meet the interaction needs of simulation components; however, the configuration of execution and time cycle modes is left to simulation designers. The scheduling of a simulation depends mostly on synchronization, as well as solvers. This is not part of the FMI standard, and the handling of fidelity problems due to interactions is left to simulation designers.
(Saidi et al. 2016) highlights the problem of correct data exchange between models, due for example to dependencies between models, using the FMI standard. In this study, the evolution of time and the way data exchange occurs is highlighted, and the focus is on the effectiveness of synchronizations. The authors propose solutions and tools to reduce execution times, based on acyclic graphs, and by proposing allocation heuristics, respecting specific constraints, in particular, interdependence between models while trying to accelerate simulations. This approach is similar to the SynDEx approach (Lavarenne et al. 1991). Our study also deals with the heuristic allocation proposal, but we do not use acyclic graphs. However, our targeted use cases for testing are frequently in real time. Simulation acceleration is not relevant.
In the position paper (Zheng and Julien 2015), the authors propose to check a CPS by observing its behavior during execution. The paper raises the problem of the current lack of control over the fidelity of the temporal behavior for a CPS simulation. It is reminded in this paper that the setting of synchronizations with FMI is left to simulation integrators, which is complicated and error-prone. The authors also focus on Modelica and assert that Modelica is not applicable to the heterogeneous models that constitute a CPS. However, (Henriksson and Elmqvist 2011) contains an example of CPS with heterogeneous models simulated with Modelica, but in this last study, the integration with tools allowing the execution time analysis is left to the future work.
In this paper, the components and architecture used are not linked to FMI, to avoid unnecessary dependencies. Nevertheless, a full description of the components is available in (Deschamps et al. 2017), and the "C-function" aspect, as well as data port, are very similar to FMI. The objective being to ensure the accuracy of the temporal behavior of a CPS simulation, as well as the expression of temporal requirements, and porting solution to FMI will not be considered before the full proof of concept.

The Simulation Distributed Architecture Model

This study is part of work on CPS simulation scheduling. The following briefly presents the method developed in (Deschamps et al. 2017) and (Deschamps et al. 2018). The method relies on the expression in a first formalism of the simulation logical architecture, sLa, and in a second formalism of the simulation execution architecture, sEa. The sLa allocation on the sEa generates the scheduling. Implementation details are added.

The Simulation Logical Architecture, sLa

The sLa allows the expression of structural and behavioral constraints of the simulation, abstracting its execution. This formalism is primarily inspired by the DEVS formalism (Zeigler et al. 2000), using components, couplings, states and internal/external functions.
The components have information about input and output ports, their refreshing frequency, the simulated system or physical phenomena states, the initial states. Internal or transition function, periodically called, update component states, and output functions update component outputs.
Channels are used to connect one component output port to another component input port.
Requirements that can be described in the sLa, for now, are:

- Latency requirements – The most common in aeronautical sector. Due to real-time data exchange

delay between avionics, data-path loops in logical time can be broken in a simulation.

- Coincidence requirements – Some simulated systems might need to receive data from other components at the same logical time to be representative.

- Affinity requirements – From simulation integrators expertise, some simulation components might operate better if in the same logical processor.

- Precedence requirements – From simulation integrators expertise, some simulation components might require to be scheduled in a specific order.

An illustration of sLa components and channels is presented in figure 3.

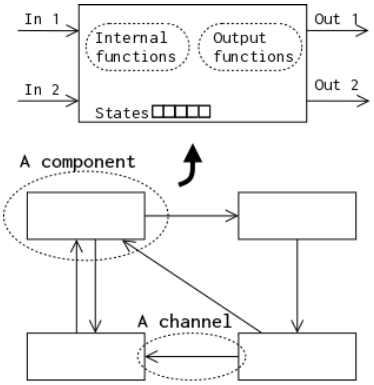


Figure 3: sLa components and channels

The Simulation Execution Architecture, sEa

The sEa is a formalism allowing the estimation of simulation execution parameters. This is a simple Architecture Description Language (ADL), in the form of a generic off-line partitioned scheduler. An illustration of the generic sEa is available in figure 4.
The sEa can represent multiple simulators, with very different execution and communication methods, as long as the following are respected:

- A global scheduler schedules logical processors, in the concurrent domain.

- Each logical processor has a similar local scheduler that schedules its tasks, in the sequential domain.

- Communication between tasks in a same logical processor and tasks in different logical processors might be different (various media, synchronizations, or latencies for instance).

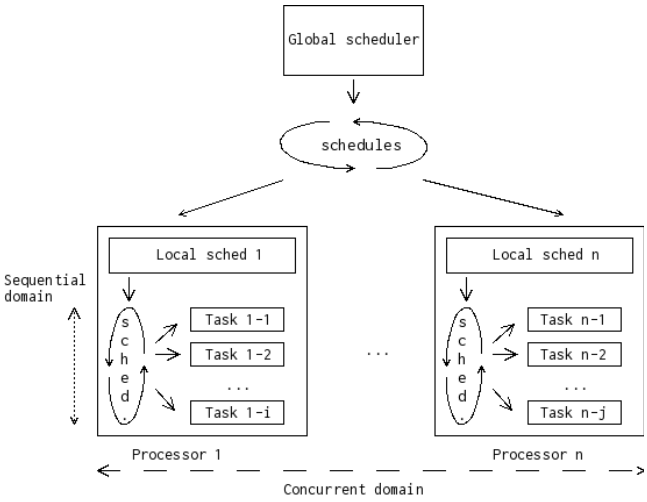


Figure 4: An sEa, with its double level of scheduling

The allocation function

The allocation function partitions the sLa components and maps them to the sEa tasks on logical processors. Knowing the sEa behavior, the resulting scheduling can be determined, and sLa requirements verified.
The allocation method is illustrated in figure 5.
The partition and mapping of components are inspired from (David et al. 1992). Partitioning consists in grouping and splitting the components into different sets. Mapping consists in choosing an order for all the sets of the chosen partition.

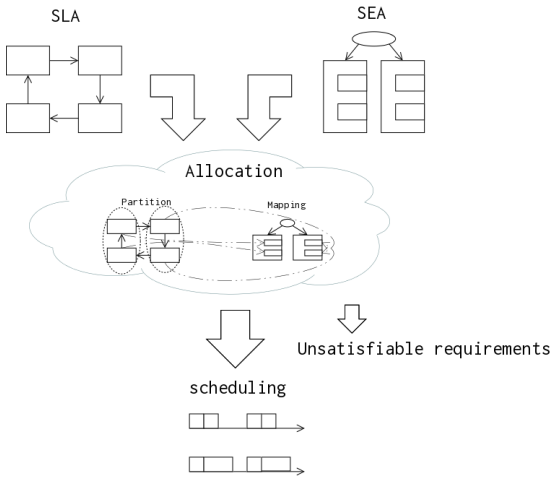


Figure 5: Determination of scheduling based on allocation of sLa on sEa

Implementation of the allocation tool

The allocation tool implements the previous formalisms and allocation. In this study, we will focus on the allocation of sLa on sEa use case. Other use cases are,

for instance, validation of existing scheduling, or verification of sLa or sEa syntax.
The allocation tool is represented in figure 6.

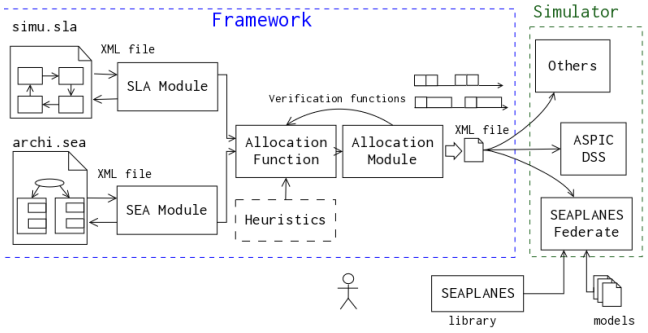


Figure 6: Allocation tool modules

The allocation tool modules

In the allocation use case, the **sLa module** and the **sEa module** read sLa and sEa respectively as XML files, and produce objects that the **allocation function** can use.
The **allocation module** contains the classes that allow the representation of scheduling, and the writing of this scheduling in an XML file. This module also contains the functions to verify the sLa requirements depending on the sEa implementation.
The problem of allocating tasks offline on a partitioned scheduler is known to be equivalent to the bin packing problem, which is NP-hard (Dhall and Liu 1978). The **allocation function** uses a **heuristic** to partition and map sLa components on sEa tasks. This **allocation function** uses the **allocation module** to create the allocation object and uses allocation object methods to verify the sLa requirements. **Heuristics** implementations are independent of the allocation function. As for now, four heuristics are implemented. The four heuristics are variations of the most known heuristics used to solve the bin packing problem. The difference with the classical heuristics being that a logical processor can schedule a component if the utilization allows it (regarding the logical processor's components' time budgets and periods), but also if the requirements are valid. If there is at least one logical processor left that is not full, but no allocation without breaking requirements, then a new allocation search is executed, considering the deletion of requirements, from the least to the most important ones. Considering an ordered set of components, and an ordered set of logical processors:

- First-fit – Each component is allocated to the first logical processor in the set. If this logical processor cannot schedule it, then the component is allocated to the next one, and so on. Logical processor set can be manipulated as a list.
- Next-fit – Same as First-fit, but the search of logical processor starts at the one following the last allocated. Logical processor set can be manipulated as a circular buffer.
- Best-fit – Search for the logical processor starts from the least, up to the most utilized one. Logical processor set can be manipulated as a binary heap, indexed by utilization.
- Worst-fit – Search for the logical processor starts from the most, down to the least utilized one. Logical processor set can be manipulated as a binary heap, indexed by utilization, in reverse order.

Compatible simulators

As long as a simulator can be described with the off-line partitioned scheduler, this simulator is compatible with the allocation results. Different simulators are already compatible with the allocation results.
ASPIC (*Atelier de Simulation Pour l'Intégration et la Conception*, in English, simulation framework for integration and design) and DSS (Distributed Simulation Scheduler) at Airbus. sEaPLANES (sEa Partition-based Logical processor Allocator Node with Extensible inline Scheduler) at ISAE-SUPAERO.
DSS is a framework for scheduling AP2633 models: Airbus simulation model containing entry points, state machine, and variables needed for scheduling. This framework schedules logical processors in logical time, and regularly synchronize data between them. Logical processors in logical time can be synchronized with wall-clock time. The primary component of a DSS simulation is its configuration file. This file contains the AP2633 models used, with their location and execution frequency. The allocation produced by the allocation module can be converted in DSS configuration file. A more detailed description of DSS can be found in (Deschamps et al. 2017).
ASPIC is a real-time simulation framework. In this framework, logical processors are scheduled according to the scheduling policy, in real-time. Depending on the scheduling policy, logical processors can be preempted. A description of ASPIC can be found in (Casteres and Ramaherirariny 2009).
sEaPLANES is a C++ framework of simulation built at the ISAE-SUPAERO in the scope of our study, based on CERTI (Bréholée and Siron 2002). CERTI is an Open source implementation of the general purpose architecture for distributed simulation HLA (High-Level Architecture) (Institute of Electrical and Electronics Engineers and IEEE-SA Standards Board 2010), developed and supported by the ONERA and the ISAE-SUPAERO. The HLA standard defines methods and a framework to build global simulation comprised of smaller simulation, the federates.

sEAPLANES is inspired from (Gervais et al. 2012), every logical processor of sEAPLANES can periodically execute tasks, exchange the data between tasks through a RunTime Infrastructure, and advance time through HLA time management services. sEAPLANES is also inspired by Airbus best practices on distributed simulation framework.
Each logical processor schedules its tasks depending on their periods, and manages intraprocessor and extraprocessor communications. Intraprocessor communication, between two tasks in a single processor, is done with shared memory. Extraprocessor communication, between two tasks in two different logical processors, is managed using the HLA publication-/subscription-based communication mechanisms. Models associated with logical processors can be manually developed, automatically generated, from instance from Matlab, or retargeted from real target or older simulations. The block diagram in fig. 8 illustrates the association of simulation models in tasks on logical processors. Fig. 7 illustrates the implementation of sEAPLANES with HLA/CERTI, with three sEAPLANES logical processors on two CPUs, for scheduling four models. Fig. 9 illustrates an allocation with associated flows of an sLA on sEAPLANES. sEAPLANES were also used in interaction with Matlab HLA toolbox-based simulation, and with Ptolemy-HLA, embedding C-code generated from Matlab (Cardoso and Siron 2018).

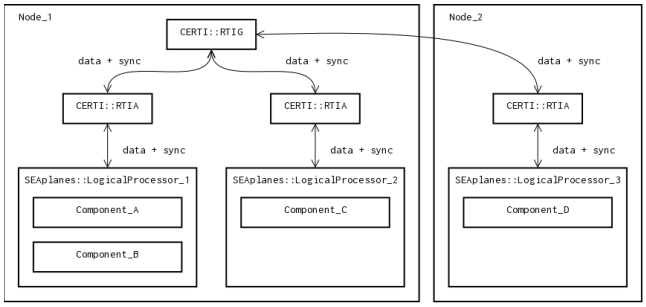


Figure 7: Example of HLA/CERTI, sEAPLANES, and R-ROSACE integration

R-ROSACE on sEAPLANES case study

This case study illustrates the allocation of an aircraft simulation, R-ROSACE, on a physical architecture of simulation, sEAPLANES.

R-ROSACE

The ROSACE (Research Open-Source Avionics and Control Engineering) case study (Pagetti et al. 2014) is a longitudinal flight controller of a medium-size aircraft. It covers different steps from the conception to the implementation of such a controller. ROSACE was chosen since it is an excellent example of a CPS, where a

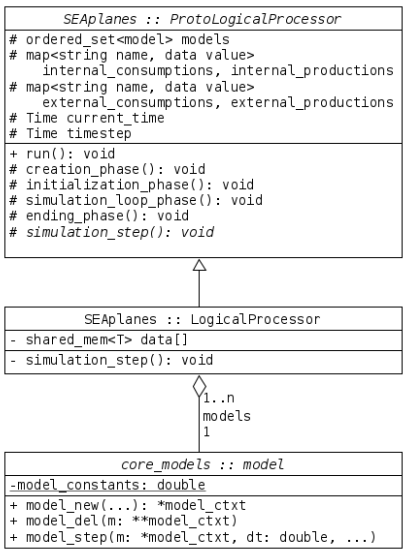


Figure 8: RROSACE sEAPLANES implementation

significant challenge is the need of interactions between the engineers responsible for the aerodynamic characteristics (physical part) and the control law (cyber part). The design of the simulation adds extra interaction with software engineers because of the need to tackle physical system requirements, such as stability, and computers science requirements, such as tasks schedulability and network resources.

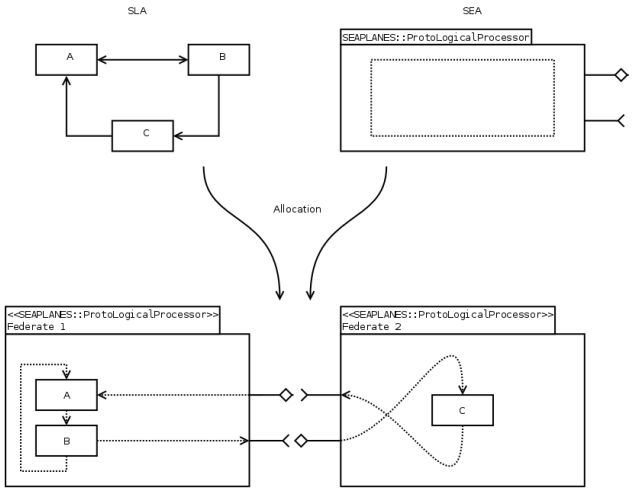


Figure 9: sLA components allocation on sEAPLANES

R-ROSACE is an extension of the open source ROSACE case study, adding redundant controllers. The component breakdown of R-ROSACE, presented in fig. 10, allows the generation of the R-ROSACE sLA. R-ROSACE has been implemented with multiple frameworks, following the architecture of simulation described in sec. "The Simulation Distributed Architecture Model"; the redundancy of controllers is illustrated in

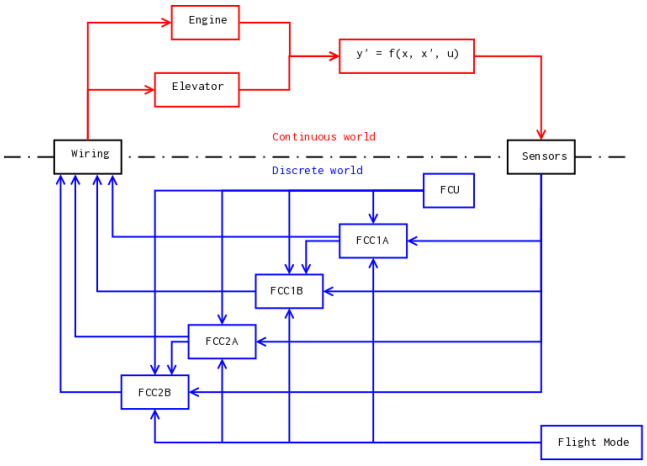


Figure 10: RROSACE components breakdown

the components breakdown figure, with FCC for Flight Control Computer. In R-ROSACE, there are 15 components, 57 channels, and 6 requirements. The requirements are due to the redundancy between FCCs. There is coincidence between each FCCs couple and the wiring, as well as between data received by FCCs couple, and FCC monitoring. An excerpt of this SLA, with a component, a channel and a requirement, is shown in lst. 1. In this excerpt, the following is written:

- Component `engine` – the engine is simulated with a period of 50 ms and an estimated time budget of 1 ms. The simulated engine component has an input port the `delta_x_c`, a change of thrust command, and an output port, `T`, a simulated thrust.
- Channel `engine T` to `flight_dynamic T` – the engine to flight dynamic thrust channel.
- Coincidence requirement between `fcc_1a`, `fcc_1b`, and `wiring` – a coincidence requirement for the wiring to receive the delta thrust computed by FCC 1A and the validation of this command by FCC 1B at the same logical time.

R-ROSACE implementations are tested through multiple operational scenarios. An operational scenario is a set of events that includes the interaction of a system with its environment and its users. A full description of R-ROSACE, operational scenarios, and Simulink models, are available at `https://svn.onera.fr/schedmcore/branches/ROSACE_CaseStudy/redundant/`. The coincidence requirements in R-ROSACE are described in (Deschamps et al. 2018).

R-ROSACE allocation on SEAPLANES

The full SEAPLANES SEA, with an unlimited number of logical processors, and without real-time constraints, is shown in lst. 2.

Listing 1: R-ROSACE SLA excerpt

```
<?xml version="1.0" ?>
<sla name="rrosace" xmlns="">
  <components>
    <component name="engine" period="50ms"
    time_budget="1ms">
      <ports_in>
        <port label="delta_x_c"/>
      </ports_in>
      <ports_out>
        <port label="T"/>
      </ports_out>
    </component>
        <!-- ... -->
  </components>
  <channels>
    <channel>
      <from component="engine" port="T"/>
      <to component="flight_dynamics" port="T"/
    >
    </channel>
        <!-- ... -->
  </channels>
  <requirements>
    <requirement weight="100">
      <coincidence>
        <path>
          <ord index="0">
            <channel>
              <from component="fcc_1a" port="
    delta_x_c"/>
              <to component="wiring" port="
    delta_x_c_1"/>
            </channel>
          </ord>
        </path>
        <path>
          <ord index="0">
            <channel>
              <from component="fcc_1a" port="
    delta_x_c"/>
              <to component="fcc_1b" port="
    delta_x_c_com"/>
            </channel>
          </ord>
          <ord index="1">
            <channel>
              <from component="fcc_1b" port="
    relay_delta_x_c"/>
              <to component="wiring" port="
    relay_delta_x_c_1"/>
            </channel>
          </ord>
        </path>
      </coincidence>
    </requirement>
        <!-- ... -->
  </requirements>
</sla>
```

The allocation tool was run with multiple configurations. Configuration consisted of different heuristics, various inputs order, and additional constraints. In all the cases tested, an allocation has been found, and few cases did not satisfy all constraints. Nevertheless, the allocation is not guaranteed, and with a different case study, we could have had configurations that do not

allow allocations. The resulting allocations have been tested with the real simulator, with tracing and data logging for verification. All the resulting scheduling estimation from allocation were similar to real schedulings on real targets.

Listing 2: sEAPLANES sEA

```
<?xml version="1.0" ?>
<sea name="seaplanes" xmlns="">
    <logical_processors multiperiodic="True"
    real-time="False">
        <intraprocessor_communication/>
        <interprocessor_communication option="
    p2p" sync="synchronous"/>
    </logical_processors>
</sea>
```

When choosing different heuristics, with the inputs provided in the default order and no additional constraints, the results are different allocations, with constraints satisfied in all the cases. First-fit and Worst-fit heuristic are similar; all the components are pushed on the first logical processor. The resulting allocation file is provided in lst. 3, with one component per logical processor, scheduled respecting their orders (`ord`), and their periods. It should be noted that the two different allocations have slightly different simulation results, all acceptable, and different performances. Simulation with first-fit/worst-fit heuristics uses 0,1 s to run one simulation second, while simulation with first-fit/worst-fit heuristics uses 0,4 s to run one simulation second. This is due to the use of the time-consuming synchronization mechanism that is increased with the second allocation.
When modifying the input orders, especially between the FCCs and wiring, or when adding constraints, such as an affinity constraint, the resulting allocation might not verify all the constraints.
Two examples can be considered:

- With the input sequence [FCC1A, wiring, FCC1B] and first-fit heuristic – After allocating FCC1A and wiring in this order on a same logical processor, which means a logical latency of 0 s between the two components, FCC1B cannot be allocated with a logical latency of 0 s from FCC1A, and a logical latency of 0 s to wiring. FCC1B is allocated in the same logical processor, and the logical latency to wiring is 50 ms. Trying to allocate FCC1B to another logical processor leads to 200 ms of logical latency from FCC1A, and 200ms to wiring.
- With the input sequence [FCC1A, FCC1B, FCC2A, FCC2B, wiring], next-fit heuristic, and adding an affinity constraint between FCC1A, FCC1B – FCC1A and FCC1B will be allocated in the same logical processor, while FCC2A and FCC2B in two different one. When trying to allocate wiring, two of the coincidence constraints cannot be satisfied, irrespective of the logical processor chosen.

Listing 3: Allocation with first-fit and worst-fit

```
<?xml version="1.0" ?>
<allocation name="alloc" xmlns="">
  <logical_processor>
    <task name="engine" ord="0" period="50ms"/>
    <task name="elevator" ord="1" period="50ms"/>
    <task name="flight_dynamics" ord="2" period="
    50ms"/>
    <task name="h_filter" ord="3" period="100ms"/>
    <task name="az_filter" ord="4" period="100ms"/
    >
    <task name="Vz_filter" ord="5" period="100ms"/
    >
    <task name="q_filter" ord="6" period="100ms"/>
    <task name="Va_filter" ord="7" period="100ms"/
    >
    <task name="fcu" ord="8" period="200ms"/>
    <task name="flight_mode" ord="9" period="200ms
    "/>
    <task name="fcc_1a" ord="10" period="200ms"/>
    <task name="fcc_1b" ord="11" period="200ms"/>
    <task name="fcc_2a" ord="12" period="200ms"/>
    <task name="fcc_2b" ord="13" period="200ms"/>
    <task name="wiring" ord="14" period="50ms"/>
  </logical_processor>
</allocation>
```

Conclusion and perspectives

In this paper, we presented an allocation tool implementing the method we developed in previous work. We developed an extensible allocation tool, that can use multiple methods for allocating sLA components on sEA tasks. We developed basic heuristics to test the allocation tool, and we displayed them with the R-ROSACE on sEAPLANES case study. Although this case study is straightforward regarding the target at Airbus, we proved on this small case study that we can automatically generate allocation and scheduling, respecting requirements, from the description of the components and execution architecture.

Nevertheless, one of the first criticisms we can make on our tool is that the implemented heuristics are too limited. Indeed, depending on the components input order, the result can vary enormously. The problem is that a component placed during the execution of one of these heuristics cannot be moved, even if a move could significantly improve the results.

Moreover, the result depends greatly on the validity of the inputs. Adding constraints that do not exist due to overzealousness can have a negative impact on the result. Conversely, not identifying a constraint can lead to unrepresentative scheduling, without our tool being able to indicate it fully.

Finally, the currently developed tool is modular and extensible, with different scheduler, communication, synchronization, and heuristics. An inconvenience is that adding a sEA with exotic local schedulers or communications is currently expensive in our tool. Many methods need to be overridden. For instance, the delays estimation on data paths. We are aware that our tool is today only a proof of concept, but such complexity is not to be neglected in the context of industrialization.

While the problems of input validities and the cost of adding new sEAs are implementation or integration challenges, for which we do not have any improvement proposals for the moment, we do have a way to improve the heuristics. The next step in our work is to implement a heuristic for which the order of the input components has no impact. We have already identified a metaheuristic candidate, simulated annealing adapted to CPS simulation scheduling, by considering simulation constraints, and generating state neighborhood from partitioning and mapping.
Finally, the step of creating sEAPLANES federates from allocation files is currently quite fast, but manual. We also plan to set up code generators from the allocation files.

Acknowledgement

The work described in this paper is supported through an Industrial Agreement for Research Training — CIFRE — financed by the National Association for Research in Technology (ANRT). This work is also financed and supervised by Airbus, and supervised by the ISAE-SUPAERO, University of Toulouse.

REFERENCES

Blockwitz T.; Otter M.; Akesson J.; Arnold M.; Clauss C.; Elmqvist H.; Friedrich M.; Junghanns A.; Mauss J.; Neumerkel D.; Olsson H.; and Viel A., 2012. *Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models*. 173–184. doi:10.3384/ecp12076173.

Bréholée B. and Siron P., 2002. *Certi: Evolutions of the onera rti prototype*. In *Fall Simulation Interoperability Workshop*.

Cardoso J. and Siron P., 2018. *Ptolemy-HLA: A Cyber-Physical System Distributed Simulation Framework. M Lohstroh et al (Eds): Lee Festschrift, Springer, LNCS 10760*, 1—-21. doi:10.1007/978-3-319-95246-8_8.

Casteres J. and Ramaherirariny T., 2009. *Aircraft integration real-time simulator modeling with AADL for architecture tradeoffs*. In *Automation Test in Europe Conference Exhibition 2009 Design*. 346–351. doi:10.1109/DATE.2009.5090686.

David V.; Fraboul C.; Rousselot J.Y.; and Siron P., 1992. *Partitioning and mapping communication graphs on a modular reconfigurable parallel architecture. Parallel Processing: CONPAR 92—VAPP V*, 43–48.

Deschamps H.; Cappello G.; Cardoso J.; and Siron P., 2017. *Toward a formalism to study the scheduling of cyber-physical systems simulations*. In *Proceedings of the 2017 IEEE/ACM 21st International Symposium on Distributed Simulation and Real Time Applications*. IEEE Computer Society, Rome, Italy.

Deschamps H.; Cappello G.; Cardoso J.; and Siron P., 2018. *Coincidence Problem in CPS Simulations: the R-ROSACE Case Study*. In *Proceedings of the 2018 9th European Congress Embedded Real Time Software and Systems*. Toulouse, France.

Dhall S.K. and Liu C.L., 1978. *On a Real-Time Scheduling Problem. Operations Research*, 26, no. 1,, 127–140.

Gervais C.; Chaudron J.B.; Siron P.; Leconte R.; and Saussié D., 2012. *Real-time distributed aircraft simulation through HLA*. In *Proceedings of the 2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications*. IEEE Computer Society, 251–254.

Henriksson D. and Elmqvist H., 2011. *Cyber-Physical Systems Modeling and Simulation with Modelica*.

Institute of Electrical and Electronics Engineers and IEEE-SA Standards Board, 2010. *IEEE standard for modeling and simulation (M & S) high level architecture (HLA): object model template (OMT) specification*. Institute of Electrical and Electronics Engineers, New York. ISBN 978-0-7381-6249-2. OCLC: 682577410.

Lavarenne C.; Seghrouchni O.; Sorel Y.; and Sorine M., 1991. *The SynDEx software environment for real-time distributed systems design and implementation*. In *European Control Conference*. vol. 2, 1684–1689.

Pagetti C.; Saussié D.; Gratia R.; Noulard E.; and Siron P., 2014. *The ROSACE case study: from Simulink specification to multi/many-core execution*. In *2014 IEEE 19th Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 309–318.

Sadvandi S.; Corbier F.; and Mevel E., 2018. *Real time and interactive co-execution platform for the validation of embedded systems*. In *Proceedings of the 2018 9th European Congress Embedded Real Time Software and Systems*. Toulouse, France.

Saidi S.E.; Pernet N.; Sorel Y.; and Khaled A.B., 2016. *Acceleration of FMU Co-Simulation On Multi-core Architectures*. 106–112. doi:10.3384/ecp16124106.

Zeigler B.P.; Praehofer H.; and Kim T.G., 2000. *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. Academic Press, San Diego, 2nd ed ed. ISBN 978-0-12-778455-7.

Zheng X. and Julien C., 2015. *Verification and Validation in Cyber Physical Systems: Research Challenges and a Way Forward*. IEEE. ISBN 978-1-4673-7088-2, 15–18. doi:10.1109/SEsCPS.2015.11.

SUPPLY CHAIN ANALYSIS AND VEHICLE SCHEDULING

VARIABLE AND CLASS-DEPENDENT SERVICE CAPACITY WITH A MULTI-CLASS ARRIVAL PROCESS

Jens Baetens
Bart Steyaert
Herwig Bruneel
Dept. of Telecommunications
and Information Processing
Ghent University
email: jens.baetens@ugent.be

Dieter Claeys
Industrial Systems Engineering (ISyE)
Flanders Make
Dept. of Industrial Systems Engineering
and Product Design
Ghent University
Belgium

KEYWORDS
Batch Service,Variable Service Capacity, N customer types

ABSTRACT

In manufacturing, a single batch server can often group a number of customers that require the same type of service. In this paper, a shared queue without customer reordering is used in order to reduce the variability of throughput time of material throughout the manufacturing process which guarantees a global First-Come-First-Served (FCFS) service discipline. This is a significant difference with the more common polling systems where each type of customer has a dedicated queue. The batch server in this paper has a variable service capacity that depends on the classes of the customers in the queue. This paper extends previous work by considering a general number of N customer classes. During the analysis, we focus on the system occupancy of this system at random slot boundaries.

INTRODUCTION

Customer differentiation has been studied mostly in the context of polling systems, for instance by Boxma et al. (2008) , Goswami et al. (2006), Dorsman et al. (2012) or Fowler et al. (2002). These types of models use a different queue for each class of customer. However, this is not always feasible because a more complicated structure is needed to filter the arrivals which increases the operational cost of the system. In this paper, we use a single shared queue for all customer classes and there is no reordering of customers allowed. This results in a global FCFS service discipline which also ensures a consistent throughput time flow of products throughout the manufacturing process. A consistent flow can be a system requirement in manufacturing for accurately predicting the delay until order completion. A global FCFS service discipline can also be used in telecommunications where strict fairness rules are required. This is described in more detail by, for instance, Avi-Itzhak and Levy (2004).

The previously mentioned papers on polling systems also incorporated batch service but they assumed the service capacity to be constant. We will look at a batch server with a stochastic capacity. Examples of these types of models can be found in Chaudhry and Chang (2004), Sikdar and Samanta (2016) or Pradhan et al. (2015). All these examples have in common that the service capacity is independent of the state of the system and

the customers in the queue. In Germs and Foreest (2013), the authors looked at a system where many of the parameters such as the service capacity and service time are dependent on the number of waiting customers.

The batch server in this paper is capable of grouping all customers of the same class but only up to the first customer of another class due to the lack of customer reordering in the shared queue. This results in a stochastic service capacity that depends on the classes of the waiting customers. Since the server can group only same-class customers, the length of such a sequence will have a key impact on the performance of this system. For this reason, we incorporate a tendency for same-class clustering in the arrival process which has been studied before for the case of two classes and without batch service in, for instance, Maertens et al. (2012). In manufacturing, this tendency often occurs due to sorting the schedule over short intervals resulting in correlation between the classes of consecutive customers.

In Baetens et al. (2016), we looked at the system occupancy of a system that combines the previously described variable capacity batch server and 2-class arrival process with clustering. The delay of this model has been studied in Baetens et al. (2018). The main contribution of our paper is extending previous work that only looked at arrival processes with 2 classes to a more generic case with N different customer classes which significantly alters the analysis. An overview of the analysed system is shown in Figure 1. The focus of the paper is the probability generating function (pgf) of the system occupancy at random slot boundaries. Differentiating between N customer classes changes the behaviour of the system since the alternation of customer classes is no longer guaranteed which significantly increases the complexity. An important part of the analysis is the proof that the denominator of the pgf of the system occupancy has N zeroes inside the unit circle which allows us to find a unique solution for all unknowns we introduced during the analysis.

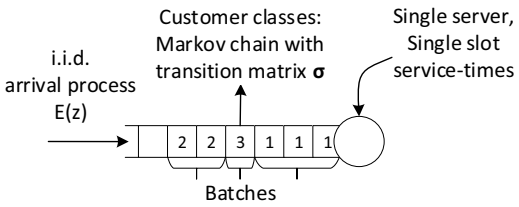


Figure 1: Overview of the system

We first give a more detailed description of the discrete-time queueing system. During the analysis of this system, we focus on the stability condition and the system occupancy at random slot boundaries. Afterwards, we present a numerical example in order to look at the impact of variance in the arrival process, and finish with drawing some conclusions.

MODEL DESCRIPTION

We start by looking at the arrival process of the discrete-time queueing system being analysed. The number of arrivals in consecutive slots is independent and identically distributed and the total number of arrivals in a single slot follows a generic distribution with the probability mass function (pmf) $e(n)$ and generating function $E(z)$ with a mean arrival rate of $\lambda = E'(1)$. We distinguish N different customer classes in the arrival stream of packets. In order to model the tendency for clustering in the arrival process, we introduce correlation between the classes of consecutive customers by using the transition matrix $\boldsymbol{\sigma}$ given by

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{1,1} & \cdots & \sigma_{N,1} \\ \vdots & \sigma_{i,j} & \cdots \\ \sigma_{1,N} & \cdots & \sigma_{N,N} \end{bmatrix},$$

where $\sigma_{i,j}$ is the probability that the class of a random customer is of class j given that its predecessor is of class i. When $\sigma_{i,i}$ is larger than the probability that a random customer belongs to class i, then the expected length of a sequence of class i customers is larger

than if there would be no correlation between consecutive customers which results in a tendency for clustering of class i customers.

As mentioned earlier, we use a single shared queue for customers of all classes and do not allow reordering of customers in this queue. The result of these restrictions is a global FCFS-service discipline. An advantage of using a shared queue is that the complexity of the system is lower than with dedicated queues resulting in a reduced cost. A disadvantage of the decreased complexity is that no optimizing of the arrival stream is possible, generally resulting in a decreased performance.

The last part of the model is the service process. The system has a single batch server that can take all same-class customers that are waiting at the head of the queue. We place no limit on the maximum service capacity but in practice the service capacity is limited by the length of a sequence of same-class customers, which follows a geometric distribution with parameter $\sigma_{i,i}$. This means that the degree of clustering, determined by the probability $\sigma_{i,i}$ will play an important role in the performance of the system. The service time of a batch of class i customers does not depend on the size of the batch, nor on the class of the customers in it, and is always equal to a single slot.

ANALYSIS

During the analysis, we will first give the system equations that capture the behaviour of the system at random slot boundaries. Then we will look at the stability condition by looking at a saturated system. At the end of the analysis, we will obtain a closed-form expression for the steady-state probability generating function of the system occupancy at random slot boundaries.

System Equations

In order to find the equations that capture the behaviour of the system, we will first define three random variables u_k, t_k and c_k. The random variable u_k represents the system occupancy at the k-th slot boundary. Because the server only groups customers that belong to the same class, we can define the type or class of a batch as the class of the customers in it. The class of the most recently initiated batch is denoted by t_k. This corresponds with the class of the batch currently being processed if $u_k > 0$ or the previous batch if $u_k = 0$. Lastly, the variable c_k corresponds to the number of customers in service during slot k which follows a geometric distribution with parameter $\sigma_{i,i}$ for a batch of class i customers limited by the system occupancy.

We can distinguish three cases in the system equations. The first case occurs when either the server is idle in the k-th slot or all customers in the system are in service, $u_k = c_k$, which results in an empty queue. If there were also no arrivals during slot k, then the system will be idle in slot $k+1$ and $t_{k+1} = t_k$. Otherwise, if there was at least one arrival during slot k, then a new service will be initiated and the class of the batch will be determined by the class of the first arrival and the system occupancy will be equal to the total number of arrivals during the k-th slot. Since the single server can only group customers belonging to the same class, the service capacity c_{k+1} follows a geometric distribution limited by the number of customers in the system. Lastly, when not all customers in the system at the k-th slot boundary belonged to the same class or $c_k < u_k$ then there will be at least one customer left behind in the queue at the next slot boundary which means a new service can be initiated and the class of the customers in this batch cannot be the same as the previous batch. Otherwise the customer(s) at the head of the queue would also have been processed in the k-th slot.

If we now assume that $t_k = i$, we obtain the

following system equations

$$(u_{k+1}, t_{k+1}, c_{k+1}) = \begin{cases} (0, i, 0) \text{ if } u_k = c_k \ \& \ e_k = 0 \ , \\ (e_k, T_{i,k}, \min(G(t_{k+1}), e_k) \\ \qquad \text{if } u_k = c_k \ \& \ e_k > 0 \ , \\ (u_k - c_k + e_k, T'_{i,k}, \min(G(t_{k+1}), u_{k+1})) \\ \qquad \text{if } 0 < c_k < u_k \ , \end{cases} \tag{1}$$

where $T_{i,k} \in \{1, \ldots, N\}$ is class j with probability $\sigma_{i,j}$, $T'_{i,k} \in \{1, \ldots, N\} \backslash \{i\}$ is of class j with probability $\sigma_{i,j}/(1 - \sigma_{i,i})$ and $G(i)$ is a geometrically distributed random variable with parameter $\sigma_{i,i}$.

Stability Condition

In order to obtain a condition under which the system is stable, we will look at a saturated system. That is a system where there are always more than enough waiting customers so that the system is never idle and the service capacity is not limited by the number of waiting customers. More details on this method can be found in Baccelli and Foss (1995). With this assumption, the system equations of Eq. are reduced to the last line since $0 < c_k < u_k$ always holds true. If we denote the steady-state probability that a random batch contains class i customers by $Pr[t = i]$, then by using the definition

$$[\boldsymbol{\sigma'}]_{i,j} := \begin{cases} 0 \text{ if } i = j, \\ \frac{\sigma_{i,j}}{1-\sigma_{i,i}} \text{ otherwise} \ , \end{cases}$$

where i and j are respectively the row and column indices, we obtain the following equations for these probabilities

$$\begin{bmatrix} Pr[t=1] \\ Pr[t=2] \\ \vdots \\ Pr[t=N] \end{bmatrix} = \boldsymbol{\sigma'} \begin{bmatrix} Pr[t=1] \\ Pr[t=2] \\ \vdots \\ Pr[t=N] \end{bmatrix} \ .$$

We obtain the probability that a random batch contains class i customers by solving this set of N equations with the additional equation for the sum of the probabilities $\sum_{i=1}^{N} Pr[t = i] = 1$.

The stability condition dictates that the mean number of customers arriving in a slot, defined in the model description as λ, must be smaller than the mean number of customer leaving. Since the size of a sequence of class i customers follows a geometric distribution with the parameter $\sigma_{i,i}$ and expected length of $\frac{1}{1-\sigma_{i,i}}$ customers, the system will only be stable if and only if the following condition holds

$$\lambda < \sum_{i=1}^{N} Pr[t = i] \frac{1}{1 - \sigma_{i,i}} \ . \tag{2}$$

System Occupancy

As mentioned earlier, we will focus on obtaining the steady-state pgf of the system occupancy at random slot boundaries. The first step is to find the probabilities that the system is idle in a random slot and the most recently initiated batch contained class i customers, denoted by $U_{I,i}$. Then we will calculate the partial pgfs $U_i(z)$, with the corresponding pmf $u_i(n) = \lim_{k\to\infty} Pr[u_k = n, t_k = i]$, of the system occupancy at random slot boundaries on which the server initiated a batch of class i customers. The probabilities $U_{I,i}$ can be calculated by using the first case in Eq. (1). We obtain that

$$\begin{aligned} U_{I,i} =& U_{I,i} E(0) + E(0) \sum_{m=1}^{\infty} u_i(m) \sigma_{i,i}^{m-1} \\ U_{I,i} =& \frac{E(0)}{1 - E(0)} \frac{U_i(\sigma_{i,i})}{\sigma_{i,i}} \ . \end{aligned} \tag{3}$$

The partial pgfs $U_i(z) = \sum_{m=1}^{\infty} u_i(m) z^m$ $(1 \leq i \leq N)$ of the system occupancy at random slot boundaries in which a batch of class i customers is initiated, are obtained by using the system equations that lead to a service initiation. These system equations correspond with the second and third case of Eq.

(1), resulting in

$$U_i(z) = \sum_{j=1}^{N} \lim_{k\to\infty} \sigma_{j,i} E[z^{e_k} | u_k = c_k, e_k > 0, t_k = j, t_{k+1} = i] + \sum_{j\neq i} \lim_{k\to\infty} \frac{\sigma_{j,i}}{1-\sigma_{j,j}} \cdot E[z^{u_{k+1}} | 0 < c_k < u_k, t_k = j, t_{k+1} = i] \ .$$

Since the service times of classes of all batches are single slots, this leads to

$$\begin{aligned} U_i(z) &= \sum_{j=1}^{N} U_{I,j}(E(z) - E(0))\sigma_{j,i} \\ &+ \sum_{j=1}^{N} \frac{U_j(\sigma_{j,j})}{\sigma_{j,j}}(E(z) - E(0))\sigma_{j,i} \\ &+ \sum_{j\neq i} \sum_{m=1}^{\infty} \sum_{n=1}^{m-1} u_j(m)\sigma_{j,j}^{n-1} z^{m-n} E(z)\sigma_{j,i} \\ &= \sum_{j=1}^{N} \frac{\sigma_{j,i} U_j(\sigma_{j,j})}{\sigma_{j,j}} \frac{E(z) - E(0)}{1 - E(0)} \\ &+ \sigma_{j,i} E(z) \sum_{j\neq i} \frac{z U_j(\sigma_{j,j}) - \sigma_{j,j} U_j(z)}{\sigma_{j,j}(\sigma_{j,j} - z)} \ . \quad (4) \end{aligned}$$

Each class i has such an equation, we obtain the following matrix equation

$$[\boldsymbol{A}]_{i,j} := \begin{cases} 1 \text{ if } i = j, \\ \frac{\sigma_{j,i} E(z)}{\sigma_{j,j} - z} \text{ otherwise} , \end{cases}$$

$$\boldsymbol{A} \cdot \begin{bmatrix} U_1(z) \\ \vdots \\ U_N(z) \end{bmatrix} = \begin{bmatrix} B_1(z) \\ \vdots \\ B_N(z) \end{bmatrix} \ ,$$

where i and j respectively correspond with the class of the new and previous service, and $B_i(z)$ is equal to

$$B_i(z) := \sum_{j=1}^{N} \frac{\sigma_{j,i} U_j(\sigma_{j,j})}{\sigma_{j,j}} \frac{E(z) - E(0)}{1 - E(0)} + \sum_{j\neq i} \frac{z\sigma_{j,i} E(z)}{\sigma_{j,j} - z} \frac{U_j(\sigma_{j,j})}{\sigma_{j,j}} \ .$$

At this point, we have a set of N equations for the N partial pgfs $U_i(z)$ which allows us to find a unique expression for each $U_i(z)$. By summing all partial pgfs $U_i(z)$ and the idle probabilities in Eq. (3), we obtain the steady-state pgf $U(z)$ of the system occupancy at random slot boundaries

$$U(z) = \sum_{i=1}^{N} U_{I,i} + \sum_{i=1}^{N} U_i(z) \ .$$

From Eq. (3) and (4), it is clear that there are still N remaining unknowns, namely $U_i(\sigma_{i,i})$ for $1 \leq i \leq N$, in this expression. The denominator $D_N(z)$ of the pgf $U(z)$ of the system occupancy at random slot boundaries is equal to the determinant of $\boldsymbol{A}$ multiplied with the term $\prod_{i=1}^{N}(z - \sigma_{i,i})$ in order to eliminate all denominators in $B_i(z)$. Due to the properties of the determinant, this is equal to multiplying each column j with $(z - \sigma_{j,j})$, which results in

$$[\boldsymbol{D}(z)]_{i,j} = \begin{cases} z - \sigma_{i,i} \text{ if } i = j, \\ -\sigma_{j,i} E(z) \text{ otherwise} , \end{cases}$$

$$D_N(z) = |\boldsymbol{D}(z)| \ .$$

In the last part of the analysis, we will prove that this denominator has N zeroes inside the circle $|z| = 1 + \epsilon$. These N zeroes will be used to obtain a set of N equations that will allow us to find the N remaining unknowns $U_i(\sigma_{i,i})$, for $1 \leq i \leq N$. This proof is based on work done by Chaudhry et al. (2016) where a similar result was obtained by using induction on the size of the matrix. For the case that N is equal to 1, we obtain $(z - \sigma_{1,1})$ for the denominator which clearly has a single zero inside the unit circle. In the next part, we assume that $D_{N-1}(z)$ has $N-1$ zeroes inside or on the unit circle and try to use this assumption to prove that $D_N(z)$ has N zeroes in the same area. We first write $D_N(z)$ as

$$D_N(z) = (z - \sigma_{N,N}) D_{N-1}(z) - \sum_{i=1}^{N-1} \sigma_{N,i} E(z) C_{N,i}(z) \ ,$$

where $C_{N,i}(z)$ corresponds with the cofactor of the element at row i and column N. By

substituting $|y_{N,i}(z)| = \frac{|C_{N,i}(z)|}{|D_{N-1}(z)|}$, we obtain

$$\left|\frac{D_N(z) - (z-\sigma_{N,N})D_{N-1}(z)}{(z-\sigma_{N,N})D_{N-1}(z)}\right| = \left|\frac{\sum_{i=1}^{N-1}\sigma_{N,i}E(z)C_{N,i}(z)}{(z-\sigma_{N,N})D_{N-1}(z)}\right| \leq \frac{\sum_{i=1}^{N-1}\sigma_{N,i}|E(z)||y_{N,i}(z)|}{|z-\sigma_{N,N}|} \ , \quad (5)$$

We can also prove that the following inequality holds for $1 \leq i \leq N$

$$|z-\sigma_{i,i}| > |\sum_{j=1,j\neq i}^{N}\sigma_{i,j}E(z)| \ . \quad (6)$$

By using the Taylor expansion, the left-hand and right-hand side of the inequality can be approximated respectively by

$$|z-\sigma_{i,i}| \geq 1-\sigma_{i,i}+\epsilon$$
$$|\sum_{j=1,j\neq i}^{N}\sigma_{i,j}E(z)| \leq (1+\lambda\epsilon+O(\epsilon^2))(1-\sigma_{i,i}) \ .$$

By trying to prove the opposite of Eq. (6), we get a result that violates the stability condition of Eq. (2) which proves that the inequality holds on the circle $|z| = 1+\epsilon$. By combining Eq. (6) with the observation that the modulus of each entry of $D_N(z)$ is less than 1, we see that the entries satisfy Hadamard's condition on the circle $|z| = 1+\epsilon$. This implies that $|y_{N,i}(z)| < 1$ for $1 \leq i < N$ which, based on Eq. (5), results in the following inequality

$$\left|\frac{D_N(z) - (z-\sigma_{N,N})D_{N-1}(z)}{(z-\sigma_{N,N})D_{N-1}(z)}\right| < \frac{\sum_{i=1}^{N-1}\sigma_{N,i}|E(z)|}{|z-\sigma_{N,N}|} < 1 \ ,$$

where we again used Eq. (6) that indicates that the modulus of the diagonal elements is larger than the sum of the modulus of the remaining elements on the row. This means that $|f(z)| > |g(z)|$, where $f(z) = (z-\sigma_{N,N})D_{N-1}(z)$ and $g(z) = D_N(z) - (z-\sigma_{N,N})D_{N-1}(z)$. Rouché's theorem, see Adan et al. (2006), then indicates that $f(z)$ and $f(z)+g(z)$ have the same number of zeroes inside the unit circle $|z| = 1$. Since we assumed at the start of the proof that $D_{N-1}(z)$ has $N-1$ zeroes and $(z-\sigma_{N,N})$ clearly has a single zero in the same area, $D_N(z) = f(z)+g(z)$ has N zeroes. These N zeroes, assuming they are all distinct, allow us to form a set of N equations to find a unique solution for the remaining unknowns. However, if not all zeroes are distinct, then it is an indication of a high degree of symmetry between two or more classes in the arrival process. This symmetry can be used to complete the set of equations necessary for finding a unique solution.

NUMERICAL EXAMPLES

During our discussion of some numerical examples, we will focus on the impact of variance in the arrival process. For this reason, we use an arrival process that is the weighted sum of two geometrics with the following pgf

$$E(z) = \frac{a}{1+\frac{\lambda}{2a}(1-z)} + \frac{1-a}{1+\frac{\lambda}{2(1-a)}(1-z)} \ ,$$

where the parameter a is chosen so that the variance in the arrival process with parameter a is equal to $\sigma_a^2 = \nu\sigma_{0.5}^2$ where $\sigma_{0.5}^2$ is the variance of the geometric distribution with mean λ. In our examples we look at three customer classes, $N = 3$, and the matrix $\boldsymbol{\sigma}$ that contains the transition probabilities, is given by

$$\boldsymbol{\sigma} = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.2 & 0.2 & 0.6 \end{bmatrix} \ .$$

In Figure 2 and 3, we look at the impact of variance in the arrival process on the mean system occupancy and the idle probability (which is the sum of all $U_{I,i}$ with $1 \leq i \leq N$) by changing the parameter ν. The parameter ν is used to compare the variance in the

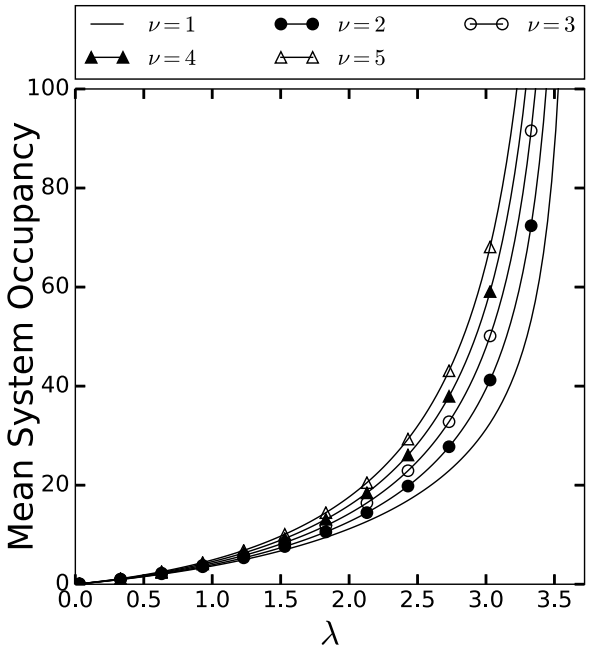


Figure 2: Impact of increased variance in the arrival process on the mean system occupancy

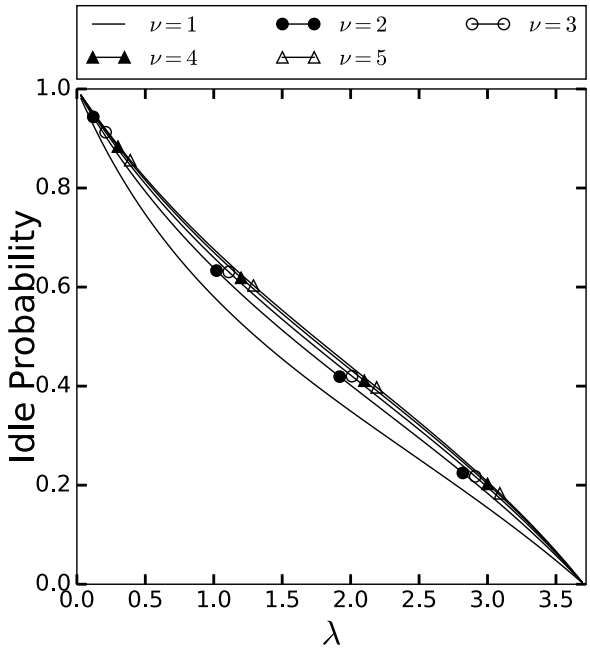


Figure 3: Impact of increased variance in the arrival process on the idle probability

arrival process to the variance of a normal geometric distribution. In Figure 2, we first notice that an increased variance in the arrival process significantly increases the mean system occupancy. One important note is that the impact of higher variance becomes stronger when the arrival rate is higher. This observation is however not true for the probability that the server is idle in a random slot. In Figure 3, we notice that although a higher variance always increases the probability that the server is idle, a point is rapidly reached at which increasing the variance further only leads to a negligible change in the idle probability. We also observe that the line for $\nu = 1$ (minimal variance) first decreases faster than the other lines but for higher arrival rates, the lines again move closer together. This means that there is an arrival rate at which the impact of the degree of variance in the arrival process reaches a maximum value and increasing the arrival rate does not further increase this impact which is different from what we observed for the mean system occupancy in Figure 2.

CONCLUSIONS

In this paper, we analysed a multi-class system with a batch server that has a service capacity that depends on the content of the queue. More specific, the batch server can group all waiting customers at the head of the queue as long as they belong to the same class. Due to the way customers are grouped together, we also incorporated a method to include clustering of same-class customers in the arrival stream. During the analysis, we looked at the condition under which the system is stable and we obtained the steady-state pgf of the system occupancy at random slot boundaries. In the numerical examples, we focused on the impact of variance in the arrival process on the idle probability and mean system occupancy. We noticed that an increase of the variance in the arrival process always leads to a significant increase of the mean system occupancy but that this is not the case when we look at the idle probability. For this performance value, a point is quickly reached at which further increasing the variance in the arrival process has a neg-

ligible impact. The main extension that we are working towards is to include a more complex service process with class-dependent service and switch-over times which would allow us to more accurately model systems where switching between types can take a long time, for instance in manufacturing.

REFERENCES

Adan I.; van Leeuwaarden J.; and Winands E., 2006. *On the application of Rouch's theorem in queueing theory. Operations Research Letters*, 34, no. 3, 355 – 360. ISSN 0167-6377. doi:https://doi.org/10.1016/j.orl.2005.05.012. URL `http://www.sciencedirect.com/science/article/pii/S0167637705000726`.

Avi-Itzhak B. and Levy H., 2004. *On measuring fairness in queues. Advances in applied probability*, 36, no. 3, 919–936.

Baccelli F. and Foss S., 1995. *On the saturation rule for the stability of queues. Journal of Applied Probability*, 32, no. 2, 494–507.

Baetens J.; Steyaert B.; Claeys D.; and Bruneel H., 2016. *System occupancy of a two-class batch-service queue with class-dependent variable server capacity.* In *International Conference on Analytical and Stochastic Modeling Techniques and Applications*. Springer, 32–44.

Baetens J.; Steyaert B.; Claeys D.; and Bruneel H., 2018. *Delay analysis of a two-class batch-service queue with class-dependent variable server capacity. Mathematical Methods of Operations Research*, 1–21.

Boxma O.; Wal J.v.d.; and Yechiali U., 2008. *Polling with batch service. Stochastic Models*, 24, no. 4, 604–625.

Chaudhry M.; Banik A.D.; Pacheco A.; and Ghosh S., 2016. *A simple analysis of system characteristics in the batch service queue with infinite-buffer and Markovian service process using the roots method: GI/C-MSP (a, b)/1/. RAIRO-Operations Research*, 50, no. 3, 519–551.

Chaudhry M.L. and Chang S.H., 2004. *Analysis of the discrete-time bulk-service queue $Geo/GY/1/N + B$. Operations Research Letters*, 32, no. 4, 355–363.

Dorsman J.; Van der Mei R.; and Winands E., 2012. *Polling systems with batch service. OR spectrum*, 34, no. 3, 743–761.

Fowler J.W.; Phojanamongkolkij N.; Cochran J.K.; and Montgomery D.C., 2002. *Optimal batching in a wafer fabrication facility using a multiproduct G/G/c model with batch processing. International Journal of Production Research*, 40, no. 2, 275–292.

Germs R. and Foreest N.V., 2013. *Analysis of finite-buffer state-dependent bulk queues. OR Spectrum*, 35(3), 563–583.

Goswami V.; Mohanty J.; and Samanta S.K., 2006. *Discrete-time bulk-service queues with accessible and non-accessible batches. Applied Mathematics and computation*, 182, no. 1, 898–906.

Maertens T.; Bruneel H.; and Walraevens J., 2012. *Effect of class clustering on delay differentiation in priority scheduling. Electronics letters*, 48, no. 10, 568–570.

Pradhan S.; Gupta U.; and Samanta S., 2015. *Queue-length distribution of a batch service queue with random capacity and batch size dependent service: $M/G_r^Y/1$. Journal of the Operational Research Society of India*, 1–15.

Sikdar K. and Samanta S., 2016. *Analysis of a finite buffer variable batch service queue with batch Markovian arrival process and servers vacation. Journal of the Operational Research Society of India*, 53, no. 3, 553–583.

SIMULATION ANALYSIS TACKLING DEMAND STOCHASTICITY THROUGH SAFETY STOCK AND LATERAL TRANSSHIPMENTS

Benedikt De Vos and Birger Raa
Ghent University, Technologiepark 903, 9052 Zwijnaarde, Belgium
email: Benedikt.DeVos@UGent.be, Birger.Raa@UGent.be

KEYWORDS
Supply Chain Collaboration, Inventory-Routing, Lateral transshipments

ABSTRACT

This paper considers a two-level distribution system consisting of a supplier and a set of retailers. The retailer demand rates are stochastic, but replenishments by the supplier are periodic, based on the known average retailer demand rates. Hence, the considered optimization problem is a Cyclic Inventory Routing Problem (CIRP). To cope with the demand variability, two safety mechanisms are introduced: safety inventory at each retailer and lateral transshipments among the retailers. The cyclic schedule for the replenishments from the supplier is generated using a state-of-the-art heuristic solution method. The safety inventory levels are determined by balancing the inventory holding costs with the expected stockout costs. For the lateral transshipments, a MILP is proposed that determines, in each period, whether redistributing items among retailers is beneficial based on their optimal inventory levels, their actual inventory levels and the costs of transport between them. To balance safety inventory holding costs with the costs of redistribution, an iterative solution approach is proposed that determines a virtual stockout cost. The proposed solution approach is analyzed in a simulation analysis under varying levels of demand variability and stockout costs.

INTRODUCTION

Increasing distribution costs have forced companies to optimize their transportation and inventory operations in order to stay profitable. Over the years, the focus of these companies has shifted from single-firm optimization to supply chain optimization, such that the overall supply chain's operations are managed in a more coordinated way and its cost efficiency is improved (Mentzer et al. 2001, Power 2005).

this paper, we consider a two-level distribution network consisting of a single supplier S that replenishes a set of retailers I with one type of product. Demand at the retailers $a_{i,t}$ (i.e., the demand at retailer i in time period t) is assumed to be stochastic with a known probability distribution. In the distribution network, there is both vertical collaboration (i.e., among players on subsequent levels of the supply chain) and horizontal collaboration (i.e., among players on the same level of the supply chain). Vertical collaboration is introduced through Vendor Managed Inventory (VMI), while horizontal collaboration is established by allowing retailers to redistribute items among each other (i.e., lateral transshipments).

Under VMI, the supplier manages the inventory at the retailers. Based on demand and inventory data, the supplier decides on the replenishment timing and quantity for all retailers. VMI thus offers the possibility to coordinate transportation and inventory decisions better and to design more cost efficient routes. The integrated optimization problem that the supplier faces is the Inventory Routing Problem (IRP). Comprehensive reviews on VMI and IRP are given by Moin and Salhi (2007), Andersson et al. (2010) and Coelho et al. (2013).

Usually, when demand is stochastic, a *reactive* solution approach is applied (as opposed to *cyclic* solution approaches when demand is constant) for the IRP. In reactive solution approaches, critical retailers are identified on a daily basis and the supplier designs new vehicle routes to replenish these critical retailers every day. However, Raa (2014) has shown that a cyclic replenishment approach is still valid under stochastic demand rates if sufficient slack is provided through safety stock at the retailers and spare capacity in the replenishment routes. A cyclic solution approach also has the advantage that replenishment frequencies and vehicle routes remain the same over time. Hence, we consider the Cyclic Inventory Routing Problem (CIRP) for the distribution between the supplier and his retailers.

To cope with the fluctuations in demand, two safety mechanisms are incorporated. First of all, safety inventory is kept at the retailers. Raa (2014) proposed a model that includes safety inventory in the CIRP solution based on the stockout cost (β_i per unit short) and inventory holding cost (η_i per unit per period) at the retailers ($i \in I$). Secondly, horizontal collaboration through lateral transshipments among the retailers is established. Retailers that have more stock on hand than necessary can redistribute items to retailers that have a high risk of running out of stock (Tagaras 1999). As a result, these lateral transshipments result in both

reduced holding costs and reduced stockouts. A literature overview on inventory models with lateral transshipments was written by Paterson et al. (2011).
The contribution of this paper lies in the combination of these two safety mechanisms and an assessment of how they influence each other. On the one hand, high levels of safety inventory lower the need for redistribution. On the other hand, since redistribution helps avoid stockouts, less safety inventory is necessary. An iterative solution approach is therefore proposed that reoptimizes the cyclic VMI delivery schedule and the safety inventory levels at the retailers based on an evaluation of how redistribution among the retailers lowers inventory and stockouts. Reiteration is done until the best balance between distribution costs, inventory costs, redistribution costs and stockout costs is achieved. A simulation analysis is performed to investigate the balance between providig safety inventory and performing redistribution among the retailers under different degrees of demand variability and under different stockout cost levels.

CIRP SOLUTION ALGORITHM

The CIRP replenishment routes are designed using a two-phase route and fleet design heuristic by Raa (2014) and Raa and Dullaert (2017). In the first phase, vehicle routes are designed using a construct-and-improve heuristic. The construction step adopts the Clarke and Wright savings heuristic (Clarke and Wright 1964), while the improvement is done by classic local search operators for routing (2-opt, relocate and exchange). Both the savings heuristic and the local search operators have to be extended for the case of cyclic inventory routing, however.
Namely, routes in the CIRP are repeated cyclically, so each route r has a cycle time T_r. that must be chosen such that the cost rates of the route is minimized. On the one hand, the supplier incurs a distribution cost F_r every time route r is made, consisting of a vehicle loading and dispatch cost, the delivery costs at the retailers replenished by route r, and the transportation cost to actually drive the route. On the other hand, the retailers incur inventory holding costs (for cycle inventory and safety inventory) and stockout costs.
Given the average daily demand rate d_i at retailer i and the cycle time T_r of the route, the customer's cycle inventory is $CI_i = \frac{d_i T_r}{2}$ (i.e., half of the delivery quantity). Given the standard deviation of a retailer's daily demand σ_i, the standard deviation of demand per cycle is $\sqrt{T_r}\sigma_i$. and the appropriate safety inventory level is $SI_i = Z_{ir}\sqrt{T_r}\sigma_i$, with Z_{ir} a safety factor depending on the length of the cycle time T_r, the inventory holding cost per unit per day η_i and the stockout cost per unit β_i. More specifically, Z_{ir} is the standard normal value corresponding to the percentile $1 - \frac{T_r \eta_i}{\beta_i}$. The expected shortage per replenishment cycle is calculated using the standard normal loss function $L(Z_{ir})$ as $ESPRC_i = L(Z_{ir})\sqrt{T_r}\sigma_i$. Hence, the total cost rate of a route TCR_r visiting retailer set I_r is:

$$TCR_r = \frac{F_r}{T_r} + T_r \sum_{i \in I_r} \frac{\eta_i d_i}{2} + \sqrt{T_r} \sum_{i \in I_r} \eta_i Z_{ir} \sigma_i + \frac{1}{\sqrt{T_r}} \sum_{i \in I_r} \beta_i L(Z_{ir}) \sigma_i \quad (1)$$

The CIRP heuristic looks for the optimal cycle time that minimizes this total cost rate of a routes by iterating over different values of T_r. However, the cycle time is restricted by the vehicle capacity κ and the storage capacity at the retailers κ_i.

REDISTRIBUTION POLICY

Even with safety inventory incorporated in the CIRP replenishments, the actual demand incurred by the retailers may lead to a stockout. The second safety mechanism to help avoid stockouts is redistribution of goods among retailers.
Every period (i.e., every day), the inventory levels of the retailers are checked and the usefulness of redistributing goods among retailers is evaluated. For every customer, in every period, a desired inventory level can be determined, based on its average daily demand and standard deviation, its inventory holding cost η_i, its stockout cost β_i, and the number of periods until the next replenishment by the supplier. This desired inventory level balances the holding costs with the expected stockout costs (and is derived similar to the safety stock level in the CIRP approach).
In every period, two types of retailers are distinguished. If the actual inventory level of a retailer is higher than its desired inventory level, this retailer is marked as a *contributor* and the difference between the actual and desired inventory level is the contributor quantity q_i. If the actual inventory level is lower than the desired inventory level, the retailer is a *receiver* and the difference between the desired and actual inventory level is the receiver quantity q_j. (Receivers are indexed with j, contributors with i.)
Both the actual and optimal inventory level correspond to a certain stockout risk before the next replenishment, and hence to a certain expected stockout cost. For receivers, the difference between the desired and actual expected stockout cost is negative and is denoted $delta_j$ (i.e., bringing a quantity q_j to receiver j results in a decreased expected stockout cost). For contributors, the difference between the desired and actual expected stockout cost is positive and is denoted $delta_i$ (i.e., using the quantity q_i of contributor i results in an increased expected stockout cost).
As soon as the set of receivers (denoted R) and the set of contributors (denoted C) have been identified, the

following optimization problem can be solved. It determines the cost of redistribution, identifies which retailers will participate and what quantities will be redistributed. In the MILP model below, x_{ij} is a binary variable denoting whether items are shipped from contributor i to receiver j, while z_{ij} is a continuous variable denoting the volume being shipped. y_i denotes whether contributor $i \in C$ is actually taking part in the redistribution.

$$\text{minimize} \quad \sum_{i \in C} \delta_i y_i + \sum_{j \in R} \delta_j y_j + \sum_{i \in C} \sum_{j \in R} c_{ij} x_{ij} \tag{2}$$

$$\text{s.t.} \quad \sum_{j \in R} z_{ij} \leq q_i y_i \qquad \forall i \in C \tag{3}$$

$$\sum_{i \in C} z_{ij} = q_j y_j \qquad \forall j \in R \tag{4}$$

$$z_{ij} \leq \min(q_i, q_j) x_{ij} \qquad \forall i \in C, \forall j \in R \tag{5}$$

$$y_i \in \{0,1\}, y_j \in \{0,1\} \qquad \forall i \in C, \forall j \in R \tag{6}$$

$$x_{ij} \in \{0,1\}, z_{ij} \geq 0 \qquad \forall i \in C, \forall j \in R \tag{7}$$

The objective function minimizes the total cost, consisting of the increased expected stockout costs of the contributors that take part in the redistribution (and for whom $\delta_i > 0$), the decreased expected stockout costs of the receivers that are involved in the redistribution (and for whom $\delta_j < 0$), and the redistribution cost. If items are shipped between a contributor and retailer, the associated cost is c_{ij}, comprising a travel cost, a loading cost at the contributor and an unloading cost at the receiver. If the objective function is negative, the decrease in expected stockout costs of the receivers outweighs the incurred redistribution cost and the increase in expected stockout cost of the contributors, which means that redistribution is beneficial in the considered time period.
Constraint (3) imposes that if contributor i is active, it can supply at most q_i, while Constraint (4) imposes that if receiver j is replenished in the redistribution, it has to receive q_j. Constraint (5) links the continuous variables z_{ij} to the binary variables x_{ij}. Note that the number of items the contributors are willing to ship to the receivers can be lower or higher than the number of items the receivers want to receive. The items can be optimally redistributed if the contributor quantity equals the receiver quantity. However, if the contributor quantity is lower than the receiver quantity, then the number of items that can be shipped to the receivers is limited to the contributor quantity.
As soon as the redistribution problem has been solved for a certain period t, the redistribution can be executed (or no redistribution happens if the model indicates that no savings can be achieved) and the inventory levels of the participating retailers are adjusted based on the values of the z_{ij} variables. Then, the regular deliveries of the cyclic replenishment scheme for that period t are made, which also affects some of the retailers' inventory levels. Finally, the retailers observe the period's demand a_{it}, after which the inventory level (and cost) as well as the stockout level (and cost) of that period can be determined. When this is simulated for a large enough number of periods, the expected costs per period of regular distribution, redistribution, inventory and stockout can be estimated.

ITERATIVE SOLUTION APPROACH

The optimal inventory level of the retailers in the CIRP replenishments is determined based on their stockout cost β_i and their inventory holding cost η_i. However, because of the redistribution that happens in some periods, the actual inventory and stockout levels (and their corresponding costs) are lower than what is anticipated in the CIRP solution, at the expense of the redistribution costs of course. Therefore, the safety inventory level of the CIRP solution should not be based solely on the given stockout cost β_i, but should also take into account the redistribution costs. Unfortunately, these are not known beforehand and can only be estimated using a simulation over many periods.
To overcome this issue, we propose the iterative solution approach shown in the flow chart of Figure 1. In this approach, the CIRP heuristic will be applied multiple times with varying stockout costs λ_i. It starts with applying the CIRP heuristic using the given stockout costs $\lambda_i = \beta_i, \forall i \in I$. This results in an initial CIRP solution that has an expected distribution cost rate DC^*, an expected inventory holding cost rate HC^* (including cycle and safety inventory) and an expected stockout cost rate SC^*. Next, the actually incurred demand rates a_{it} are generated for all retailers over a long enough planning horizon P and the simulation is performed that evaluates the redistribution potential in each period. Once this is done for the whole planning horizon, the incurred redistribution cost rate RC, the inventory holding cost rate HC and stockout cost rate SC are estimated.
Due to the redistribution, instead of a stockout cost rate SC^* (that the CIRP solution anticipates), an actual redistribution plus stockout cost rate $RC + SC$ is incurred. Thus, the CIRP heuristic overestimates the stockout cost rate. Therefore, in the next iteration, the stockout costs λ_i are updated based on the ratio of the actual cost rates and the anticipated cost rates as follows.

$$Ratio = \frac{SC + RC}{SC*} \tag{8}$$

$$\lambda_i = \lambda_i \cdot Ratio \tag{9}$$

The CIRP heuristic is then reapplied using the adjusted stockout costs λ_i, resulting in a new replenishment solution and new expected cost rates. The simulation that considers redistribution is also repeated, resulting in a

new redistribution cost rate RC, inventory holding cost rate HC and stockout cost rate SC. Note that the stockout cost rate SC is calculated with the original β_i values and not with the virtual λ_i values. This gives a new ratio of actual vs. anticipated cost rates, so the stockout cost rates λ_i can be readjusted and the cycle can be repeated. When the anticipated stockout cost rate SC^* (using the artificial λ_i values) is lower than the actual cost rate $RC + SC$ (using the original β_i values), the λ_i values will increase in the next iteration. That next iteration is the final iteration. The iteration that resulted in the lowest total cost rate TC (i.e., the sum of the distribution cost rate, the redistribution cost rate, the inventory holding cost rate and the stockout cost rate) is returned as the final solution. The optimal total cost rate is denoted by TC_{opt}.

SIMULATION ANALYSIS

To test the proposed solution approach, a simulation analysis is performed. The solution approach was applied on two datasets containing 86 and 120 retailers. The location of the retailers are generated randomly in such a way that they are uniformly distributed in a circle around the supplier. The distances between the supplier and retailers are assumed to be the Euclidean distances. The vehicles used by the supplier for the CIRP distribution have a capacity $\kappa = 100$ units. The storage capacity at the retailers is assumed to be high enough to store a full truckload (i.e. $\kappa_i = 100$ for all retailers). The transportation cost is set at 1.2 euro per km, the vehicle loading cost is 20 euro per driven tour and the unloading cost is 10 euro at each retailer. The stockout cost at the retailers has two levels, high (i.e., $\beta_i = 200$ euro per unit per day) or low (i.e., $\beta_i = 20$ euro per unit per day).

The daily demand quantities at the retailers a_{it} are generated using a truncated normal distribution with mean d_i. The solution approach is applied under varying levels of demand variability. The coefficient of variation CV (i.e., the ratio of standard deviation to mean) was increased stepwise from 10 to 50% for all retailers.

Table 1 shows the initial results for the different demand variability levels and stockout costs. The safety inventory level in the CIRP and the redistribution policy in this step are based on the stockout costs β_i. It first shows the expected total cost rate TC^*, the expected distribution cost rate DC^*, the expected cycle inventory holding cost rate CHC^*, the expected safety inventory holding cost rate SHC^* and the expected stockout cost rate SC^* calculated by the CIRP heuristic. The following columns then show the simulated total cost rate TC, holding cost rate HC, stockout cost rate SC and redistribution cost rate RC. The distribution cost rate DC from the simulation is not shown, since the cyclic replenishments are always executed as planned (i.e., $DC = DC^*$.) The *Ratio* and resulting adjusted stockout cost λ_i are in the final two columns. In all cases, it can be seen that the λ_i for the second iteration is smaller than the initial values β_i.

Looking at the results in Table 1, some observations can be made. First of all, the CIRP heuristic copes well with demand variability as it comes up with different solutions for varying levels of variability or varying stockout costs, in which the expected costs of distribution, inventory holding and stockout are carefully balanced. Second, safety inventory as well as stockout cost rates increase with demand variability. Third, and importantly, including the redistribution option allows for a decrease in the total cost rate. The savings that can be made in stockout costs by redistribution are indeed often larger than the costs that the redistribution incurs. As can be expected, this savings increases both with the level of variability and with the magnitude of the stockout cost.

Table 2 shows the results of the second iteration of both instances. It can be seen that *Ratio* is larger than 1 in nine of the cases, meaning that the third iteration will already be the final iteration for those. For the other cases, further iterations are necessary as long as $Ratio < 1$. For five cases, a fourth iteration is required, another five cases require a fifth iteration, and one case needs six iterations. Table 4 on the final page shows the results of these additional iterations.

As long as $Ratio < 1$, updating λ_i leads to a decrease of this artificial stockout cost. In the consecutive CIRP solution, this decreased stockout cost results in lower safety inventory levels and thus lower inventory holding costs at the retailers. As the safety inventory levels given by the CIRP solution decrease over the iterations, we can also see in the simulation results that the redistribution costs increase. This is due to the fact that with lower safety inventory levels, stockout risks are higher and thus more redistribution is required to reduce stockout costs. Overall, across the iterations, the solution that results in the best balance between distribution costs, holding costs, redistribution costs and stockout costs can then be identified. For dataset 1 with $CV = 0.1$ and $\beta_i = 20$ for example, three iterations were performed. Their respective simulated total cost rates are 2098.4, 2093.0 and 2091.3. Hence, the optimal solution is found in the third iteration and corresponds to the CIRP solution obtained with $\lambda_i = 19.8$.

To quantify the savings potential that redistribution offers, these optimal cost rates are compared to the cost rates of the base scenario TC_b. These are obtained by running the simulation on the initial CIRP solution (still using the same daily demand values a_{it}), but without allowing redistribution. Note that these base scenario cost rates are not necessarily equal to the expected cost rates of the CIRP solution due to variability in demand. In Table 3, the optimal total cost rates TC_{opt}, the base scenario cost rates TC_b and the savings potential are shown for all cases. It can be seen that the solution approach generates savings for the overall distribution network.

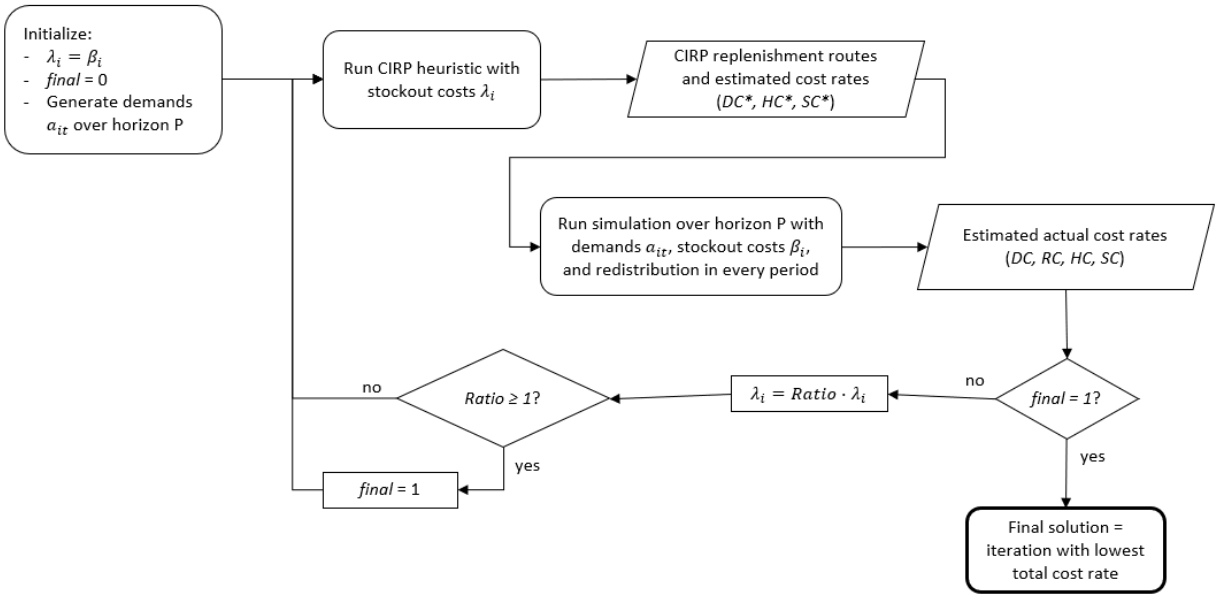


Figure 1: Flow chart iterative solution approach

Table 1: Expected vs. simulated cost rates in initial iteration

			CIRP solution					Simulation					
I	CV	β_i	TC^*	DC^*	CHC^*	SHC^*	SC^*	TC	HC	SC	RC	$Ratio$	new λ_i
1	0.1	20	2098.8	1304.0	701.1	65.8	27.8	2098.4	767.0	27.3	0.1	0.984	19.7
1	0.2	20	2195.8	1335.7	674.0	131.8	54.3	2186.6	809.2	40.6	1.1	0.768	15.4
1	0.3	20	2304.5	1373.1	654.5	196.9	80.0	2295.9	853.2	64.2	5.4	0.869	17.4
1	0.4	20	2410.6	1397.1	647.0	261.4	105.1	2392.5	913.0	70.2	12.2	0.783	15.7
1	0.5	20	2535.6	1394.6	680.5	325.6	134.9	2509.1	1009.6	82.1	22.8	0.777	15.5
1	0.1	200	2164.9	1338.6	671.8	132.8	21.7	2160.2	804.5	16.7	0.4	0.787	157.4
1	0.2	200	2344.5	1392.2	646.4	263.4	42.5	2333.3	910.3	28.3	2.5	0.724	144.9
1	0.3	200	2529.9	1446.7	629.8	390.7	62.7	2498.9	1020.7	27.0	4.5	0.503	100.5
1	0.4	200	2720.4	1444.9	661.8	527.5	86.1	2674.0	1191.7	24.7	12.6	0.434	86.8
1	0.5	200	2904.6	1481.0	655.4	660.6	107.6	2843.2	1317.1	31.1	14.1	0.420	84.0
2	0.1	20	3175.8	2130.5	897.4	107.5	40.4	3173.6	1005.8	37.0	0.2	0.923	18.5
2	0.2	20	3357.5	2151.3	910.0	215.0	81.1	3347.4	1126.0	67.3	2.8	0.865	17.3
2	0.3	20	3550.9	2204.7	904.2	321.4	120.5	3536.0	1228.3	93.6	9.4	0.855	17.1
2	0.4	20	3721.1	2234.1	896.5	429.7	160.8	3689.4	1327.9	103.7	23.7	0.792	15.8
2	0.5	20	3907.5	2291.9	883.3	534.0	198.3	3858.4	1421.4	115.1	30.1	0.732	14.6
2	0.1	200	3299.5	2181.0	882.5	204.5	31.5	3290.1	1087.3	20.2	1.7	0.693	138.6
2	0.2	200	3608.7	2269.6	870.8	406.0	62.3	3595.6	1277.5	41.6	6.9	0.778	155.7
2	0.3	200	3901.6	2331.9	866.5	609.6	93.5	3858.1	1475.6	40.0	10.6	0.541	108.1
2	0.4	200	4211.9	2375.3	892.2	818.1	126.3	4143.6	1707.0	48.5	12.8	0.485	97.1
2	0.5	200	4438.3	2466.9	826.1	994.6	150.7	4362.7	1820.4	62.3	13.2	0.501	100.1

These collaborative savings appear to increase as the coefficient of variation increases. Furthermore, the savings are also higher in the datasets with high stockout costs at the retailers (i.e., $\beta_i = 200$) than in the datasets with low stockout costs at the retailers (i.e., $\beta_i = 20$).

CONCLUSION

In this paper, the benefits of two safety mechanisms in transportation in a two-stage distribution network with stochastic demand are studied. Namely, safety stock in the supplier replenishment routes and redistribution of inventory among retailers.

The supplier replenishes the retailers in a VMI setting. Although demand is stochastic, replenishments by the supplier are cyclic. The supplier distribution routes and costs are determined using a state-of-the-art CIRP solution heuristic. The safety inventory levels that are included in the CIRP replenishments are based on the optimal inventory levels of the retailers (which are de-

Table 2: Expected vs. simulated cost rates in second iteration

			CIRP solution					Simulation					
I	CV	λ_i	TC^*	DC^*	CHC^*	SHC^*	SC^*	TC	HC	SC	RC	$Ratio$	new λ_i
1	0.1	19.7	2093.5	1330.0	670.5	65.5	27.1	2093.0	735.8	27.1	0.1	1.004	19.8
1	0.2	15.4	2194.1	1338.6	671.8	113.3	54.1	2190.3	787.1	62.2	2.4	1.195	18.4
1	0.3	17.4	2307.5	1339.8	692.7	180.3	82.3	2301.9	877.1	76.4	8.7	1.033	18.0
1	0.4	15.7	2404.5	1378.8	659.8	228.8	107.4	2376.6	897.0	84.6	16.2	0.939	14.7
1	0.5	15.5	2523.1	1416.5	649.7	286.1	132.7	2507.0	944.5	112.4	33.5	1.100	17.1
1	0.1	157.4	2165.5	1338.6	671.8	126.4	22.6	2164.4	798.9	24.7	2.2	1.194	187.9
1	0.2	145.0	2346.8	1406.9	634.9	244.8	43.6	2335.2	878.8	41.6	7.9	1.134	164.3
1	0.3	100.5	2539.6	1414.7	646.6	340.3	69.4	2472.2	988.9	53.9	14.7	0.989	99.4
1	0.4	86.8	2747.2	1412.2	672.5	442.7	95.4	2622.6	1120.5	62.9	27.0	0.942	81.7
1	0.5	84.0	2926.7	1456.5	649.4	543.5	116.5	2766.8	1197.2	76.6	36.5	0.971	81.5
2	0.1	18.5	3172.1	2118.0	906.5	103.6	40.6	3167.9	1009.9	39.8	0.1	0.983	18.1
2	0.2	17.3	3351.7	2146.2	910.2	201.0	81.6	3347.6	1113.2	84.4	3.9	1.082	18.7
2	0.3	17.1	3535.7	2164.6	928.0	296.8	125.0	3515.4	1226.5	111.1	13.1	0.993	17.0
2	0.4	15.8	3719.1	2212.2	918.6	380.8	164.5	3673.9	1304.4	128.0	29.3	0.957	15.2
2	0.5	14.6	3897.3	2226.7	929.4	455.8	208.9	3842.8	1388.3	163.7	64.0	1.090	16.0
2	0.1	138.6	3280.3	2149.8	891.9	190.6	33.2	3275.4	1083.5	39.8	2.4	1.269	175.8
2	0.2	155.7	3592.3	2236.6	881.7	390.6	65.0	3566.4	1269.9	52.6	7.2	0.921	143.4
2	0.3	108.1	3919.8	2303.3	885.0	542.3	102.3	3823.2	1431.2	68.7	20.1	0.868	93.8
2	0.4	97.1	4225.5	2343.6	888.1	710.6	137.5	4080.7	1602.7	89.3	45.1	0.977	94.9
2	0.5	100.1	4538.4	2347.4	935.1	902.1	177.1	4315.3	1833.8	76.7	57.4	0.758	75.9

Table 3: Savings potential of redistribution

I	CV	β_i	TC_b	TC_{opt}	Saving %
1	0.1	20	2099.92	2091.33	0.41%
1	0.2	20	2196.74	2185.59	0.51%
1	0.3	20	2306.49	2291.07	0.67%
1	0.4	20	2419.66	2376.59	1.78%
1	0.5	20	2550.02	2496.41	2.10%
1	0.1	200	2161.43	2155.70	0.27%
1	0.2	200	2347.84	2320.78	1.15%
1	0.3	200	2534.46	2472.17	2.48%
1	0.4	200	2717.03	2622.56	3.48%
1	0.5	200	2910.76	2762.31	5.10%
2	0.1	20	3176.70	3167.77	0.28%
2	0.2	20	3360.65	3347.43	0.39%
2	0.3	20	3562.26	3510.88	1.44%
2	0.4	20	3722.12	3669.86	1.40%
2	0.5	20	3911.39	3834.15	1.97%
2	0.1	200	3295.16	3275.43	0.60%
2	0.2	200	3611.66	3563.52	1.33%
2	0.3	200	3900.58	3790.83	2.81%
2	0.4	200	4205.33	4018.99	4.43%
2	0.5	200	4442.44	4273.94	3.79%

termined using their inventory holding cost and their stockout cost). Despite replenishments being cyclic, not all retailers are replenished with the same cycle time. Hence, they do not receive deliveries on the same days. As a consequence, redistribution of inventory among retailers can take place in each time period within the considered planning horizon. Redistribution among the retailers brings along an additional transportation cost, namely the redistribution cost. The redistributed quantities are determined by an optimization model that determines the optimal redistribution cost and compares it to the difference in expected stockout cost among the retailers. When the optimization model determines that the saving in expected stockout cost due to redistribution outweighs the cost of redistribution, redistribution is beneficial and the items are redistributed.

In order to find an appropriate balance between the two safety mechanisms (safety inventory included in the CIRP and redistribution among the retailers), an iterative solution approach is developed. It starts by performing the CIRP based on the given stockout costs at the retailers. In the next iterations, a virtual stockout cost is determined based on the cost of redistribution and the stockout cost from the previous iteration. The CIRP heuristic and the optimization of redistribution are reapplied until the virtual stockout cost has converged. The redistribution creates savings through reductions in both stockout and inventory costs. On the other hand, redistribution also brings along an extra transportation costs. The additional redistribution cost has to be balanced with the reduction in holding and stockout costs. The optimal solution can be identified as the iteration resulting in the lowest total cost rate.

A simulation analysis is performed to test the proposed solution approach. Tests on two datasets under varying levels of demand variability and stockout costs have shown that the solution approach entails benefits for the distribution network. The savings differ according to the level of demand variability and stockout cost. Higher levels of demand variability and higher stockout costs lead to higher savings. Furthermore, we can see that over the iterations in the iterative solution approach, inventory holding costs decrease as the virtual stock-

out cost decreases, due to lower safety inventory levels. These lower safety inventory levels also bring along an increase in the redistribution costs though.
More in-depth research into larger datasets is necessary. A design of experiments should be performed to further test the solution approach under varying problem characteristics, like changing inventory holding costs and transportation cost. Furthermore, we need to consider that the retailers only want to participate in the collaboration when they receive a fair share of the savings. Hence, we need to investigate how the savings can be shared over all participants in a fair way.

REFERENCES

Andersson H.; Hoff A.; Christiansen M.; Hasle G.; and Løkketangen A., 2010. *Industrial aspects and literature survey: Combined inventory management and routing. Computers & Operations Research*, 37, no. 9, 1515–1536.

Clarke G. and Wright J.W., 1964. *Scheduling of vehicles from a central depot to a number of delivery points. Operations research*, 12, no. 4, 568–581.

Coelho L.C.; Cordeau J.F.; and Laporte G., 2013. *Thirty years of inventory routing. Transportation Science*, 48, no. 1, 1–19.

Mentzer J.T.; DeWitt W.; Keebler J.S.; Min S.; Nix N.W.; Smith C.D.; and Zacharia Z.G., 2001. *Defining supply chain management. Journal of Business logistics*, 22, no. 2, 1–25.

Moin N.H. and Salhi S., 2007. *Inventory routing problems: a logistical overview. Journal of the Operational Research Society*, 58, no. 9, 1185–1194.

Paterson C.; Kiesmüller G.; Teunter R.; and Glazebrook K., 2011. *Inventory models with lateral transshipments: A review. European Journal of Operational Research*, 210, no. 2, 125–136.

Power D., 2005. *Supply chain management integration and implementation: a literature review. Supply chain management: an International journal*, 10, no. 4, 252–263.

Raa B., 2014. *Cyclic versus reactive planning for inventory routing. Procedia-Social and Behavioral Sciences*, 111, 909–917.

Raa B. and Dullaert W., 2017. *Route and fleet design for cyclic inventory routing. European Journal of Operational Research*, 256, no. 2, 404–411.

Tagaras G., 1999. *Pooling in multi-location periodic inventory distribution systems. Omega*, 27, no. 1, 39–59.

Table 4: Expected vs. simulated cost rates in consecutive iterations

			CIRP solution					Simulation					
I	CV	λ_i	TC^*	DC^*	CHC^*	SHC^*	SC^*	TC	HC	SC	RC	$Ratio$	new λ_i
Iteration 3													
1	0.1	19.8	2093.5	1330.0	670.5	65.5	27.2	2091.3	736.7	24.6	0.1	0.908	-
1	0.2	18.4	2195.0	1335.7	674.0	125.8	54.6	2185.6	801.1	46.4	2.4	0.893	-
1	0.3	18.0	2307.4	1339.9	692.2	184.2	81.8	2291.1	879.4	63.3	8.5	0.878	-
1	0.4	14.7	2406.4	1384.5	657.1	219.5	106.8	2387.9	882.0	101.2	20.1	1.137	16.7
1	0.5	17.1	2519.4	1420.0	642.5	300.4	133.8	2496.4	947.0	101.6	27.8	0.968	-
1	0.1	187.9	2161.5	1356.6	651.0	130.7	21.8	2155.7	781.9	15.9	1.2	0.788	-
1	0.2	164.3	2347.6	1392.4	649.5	252.4	43.7	2320.8	903.3	20.9	4.3	0.575	-
1	0.3	99.4	2542.6	1417.1	649.3	338.3	68.6	2480.6	989.1	51.2	23.2	1.086	108.0
1	0.4	81.7	2755.5	1442.2	652.4	431.6	93.7	2637.6	1086.8	73.4	35.2	1.159	94.8
1	0.5	81.5	2932.4	1456.5	649.4	539.4	117.1	2765.8	1196.6	74.8	37.9	0.963	78.5
2	0.1	18.1	3168.2	2162.8	860.2	102.0	39.2	3167.8	961.6	43.4	0.002	1.108	20.1
2	0.2	18.7	3362.5	2150.7	915.4	208.9	81.9	3353.8	1124.5	76.2	2.5	0.960	-
2	0.3	17.0	3534.7	2149.9	940.1	297.4	125.1	3512.3	1237.5	111.0	13.9	0.998	17.0
2	0.4	15.2	3717.0	2187.0	938.3	371.1	167.2	3669.9	1313.6	135.2	34.1	1.012	15.3
2	0.5	16.0	3890.6	2257.8	898.2	480.0	203.2	3834.2	1384.5	142.8	49.1	0.944	-
2	0.1	175.8	3284.9	2158.9	889.2	200.0	32.3	3283.0	1090.1	31.6	2.4	1.052	-
2	0.2	143.4	3598.5	2253.6	871.2	382.8	65.2	3563.5	1254.2	49.2	6.5	0.856	122.6
2	0.3	93.8	3917.5	2283.2	886.9	527.8	103.0	3790.8	1415.7	68.9	23.0	0.892	83.7
2	0.4	94.9	4249.1	2360.2	894.8	704.0	137.6	4062.4	1600.3	71.5	30.4	0.741	70.3
2	0.5	75.9	4580.0	2354.8	912.5	842.6	178.4	4273.9	1761.5	98.5	59.1	0.884	67.0
Iteration 4													
1	0.4	16.7	2407.4	1375.0	668.6	236.0	106.7	2396.7	911.6	92.1	18.0	1.032	-
1	0.3	108	2534.7	1413.2	647.8	346.3	68.8	2474.1	994.6	45.8	20.5 0.964	-	
1	0.4	94.8	2740.4	1446.0	653.6	446.3	92.2	2640.1	1104.0	61.0	29.1	0.977	-
1	0.5	78.5	2939.5	1457.5	649.9	533.2	117.3	2769.8	1187.0	70.9	54.3	1.067	83.8
2	0.1	20.1	3176.7	2131.4	897.9	107.5	40.0	3174.0	1005.5	37.0	0.1 0.927	-	
2	0.3	17	3537.8	2176.2	918.7	297.6	123.5	3515.2	1217.8	108.2	13.0	0.981	16.7
2	0.4	15.3	3717.0	2206.6	922.5	374.1	163.6	3670.8	1299.0	129.8	35.4	1.010	-
2	0.2	122.6	3604.5	2213.0	906.3	375.3	67.4	3566.1	1282.6	57.6	13.0	1.047	128.4
2	0.3	83.7	3932.6	2265.7	895.4	518.3	106.0	3801.1	1413.5	90.9	31.0	1.150	96.3
2	0.4	70.3	4259.3	2339.6	868.6	651.7	140.4	4019.0	1523.5	103.1	52.7	1.110	78.0
2	0.5	67	4629.9	2358.7	912.9	813.1	182.6	4283.4	1730.2	121.0	73.5	1.065	71.3
Iteration 5													
1	0.5	83.8	2927.5	1456.5	649.4	542.8	116.9	2762.3	1195.6	70.5	39.7	0.94	-
2	0.3	16.7	3544.8	2209.2	897.3	293.8	120.7	3527.9	1194.3	111.8	12.5	1.03	17.2
2	0.2	128.4	3591.7	2240.9	873.4	374.2	66.3	3568.5	1247.0	66.5	14.1	1.22	-
2	0.3	96.3	3926.3	2296.0	884.6	531.2	103.3	3840.5	1416.6	97.4	30.6	1.24	-
2	0.4	78	4241.1	2360.1	864.1	666.0	136.9	4038.9	1532.2	106.4	40.2	1.07	-
2	0.5	71.3	4541.7	2425.8	834.1	804.7	170.1	4289.8	1644.0	142.0	78.1	1.29	-
Iteration 6													
2	0.3	17.2	3531.5	2179.9	910.5	299.1	122.1	3510.9	1213.5	106.6	11.0	0.962	-

Bi-directional Service Strategy for Reusable Transportation Packaging (RTPs) Pooling System

Di Zhang and Uwe Clausen
Fakultät Maschinenbau / Institut für Transportlogistik
Leonhard-Euler-Str. 2
D-44227 Dortmund, Germany
Email:{zhang@itl.tu-dortmund.de}

KEYWORDS

Reusable transport packaging; pooling system; closed-loop supply chain; resource utilization; service strategy

ABSTRACT

Issues of logistics strategy and organization have been discussed in the literature for at least 50 years. Unlike the cost issue which is the primary goal in general logistics system optimization, for RTPs in a precise planning period with a stable service capacity, the faster the turnover is, i) the higher the service level; ii) the stronger the market competitiveness; iii) the better the profitability of enterprise has. This paper proposes a bidirectional service strategy for RTPs pooling system and simulates strategies in different scenarios.

INTRODUCTION

Supply chain management is defined as 'an integrative philosophy to manage the total flow of a distribution channel from the supplier to the ultimate user' (Ellram, L.M. and Cooper, M.C., 1990). This means greater coordination of business processes and activities, such as inventory management and transportation management, across the entire channel and rather than just between a few channel pairs (Cooper, M.C. and Ellram, L.M., 1993). To adapt to the flexibility of the supply chain, companies should develop and implement a formal logistics strategy. When a company creates a logistics strategy, it is defining the service levels at which its logistics organization is most cost effective (Murray, 2017).

Furthermore, supply chains are constantly changing and evolving, a company may develop a number of logistics strategies for specific product lines, specific countries, or specific customers. Some papers note that logistics strategy is the set of guiding principles, driving forces and ingrained attitudes that help to coordinate goals, plans and policies between partners across a given supply chain. As logistics plays an important role in supporting a company's products in the market place, the implications of this 'full role' extend far beyond the boundaries of the company itself (Web reference 1). Challenges for the 3PL are described as to balance between the ability of adapting highly to individual customers and organizing the systems and the business for coordination of several customers. The way this is balanced will guide the strategic development of the 3PL providers and is of vital importance to the resources needed, activities to be performed, and core competence development (Hertz and Alfredsson, 2003).

During the 1980s, the literature on logistics began to examine alternative perspectives on logistics organization and logistics strategy. Issues of logistics strategy and organization have been discussed in the literature for at least 50 years. Researchers determine the strategic objectives: defining the service levels at which its logistics organization is most cost effective; propose three strategic orientations: process strategy, market strategy and information strategy; demonstrate the impact of logistics strategy on customer service level; and put forward four distinct levels of logistics organization: strategic, structural, functional and implementation and validate their conclusions in several industries and variety of modes of transport (McGinnis and Kohn, 1990, 2002, Cooper and Ellram, 1993, Porter, 1996, Ballou, 1997, Hertz and Alfredsson, 2003, Lee, Kuoliang., 2009, Bolboaca, Codruta., 2010, Liu et al., 2015, Rudolf and Douglas M, 2016, MurrayMiao et al., 2017).

With the development and practice of environmental theory, reusable articles are widely used in logistics industry. This paper introduces a bidirectional service strategy for reusable transportation packaging (RTPs) pooling system. Section 2 introduces the change in research perspective; Section 3 explains the content of bidirectional service strategy and critical data calculation model is proposed in section 4. Section 5 simulates a case and discusses the results. The final section presents the main conclusions and extension of the research.

CHANGE IN RESEARCH PERSPECTIVE

For RTPs pooling system, the reusable nature of RTPs makes CLSC of reusable articles different from end-of-use CLSC, used and brand new products are perfect substitutes and serve the same markets (Garrasco-Gallego 2010). From a depot the sender is provided with the number of RTPs he needs. After these have been transported to the recipient, the

recipient bundles the empty containers and stores them until a sufficient number has been accumulated for cost-effective collection (Kroon and Vrigens, 1994). Throughout the process, if the depot has sufficient inventory and has strong service capacity, the RTPs owners have more confidence and motivation to get new contract. New demand generates the same amount of collection needs in reverse channel. Fast and efficient collection activities guarantee that the depot has sufficient inventory which means the supply capacity of depot is reliable (Zhang and Clausen, 2016).

Unlike the cost issue which is the primary goal in general logistics system optimization, for RTPs in a certain planning period with a stable service capacity, the faster the turnover means, i) the higher the service level; ii) the stronger the market competitiveness; iii) the better the profitability of enterprise has. By the same token, the efficient use of other resources which involved in the RTPs pooling system also plays a positive role in optimizing network efficiency. When the market demand exceeds the maximum service capacity of the network, managers increase resource input according to the demand growth rate in order to maintain a high level of service.

The reusable feature of RTPs brings some benefits as well as some new changes to the logistics managers. Managers who deal with reusable elements have reported difficulties in orchestrating these logistics systems when they interact with Garrasco-Gallego (2010). These operational challenges are also remarked in the existing literature, where several authors point out that managing reusable articles system can become trickier than expected (McKerrow, 1996, Twede and Clarke, 2005). In last decades, researchers pay more attention to reusable articles, topics such as closed-loop inventory routing problem (Soysal 2016), scheduling problems with time window (Mensendiek, 2015), carbon emission costs and management policies (Sarkar et al, 2017) have been studied.

Some new elements are added to the traditional logistics strategy triangle (in Figure.1): delivery and collection rules/methods added to inventory strategy; flow balance considered in the location strategy; and carbon emissions issues are being addressed in transportation strategy. In the general strategy triangle, inventory strategy concerns product purchase and supply scheduling plan, forecasting inventory level and making decisions. The new focus of inventory strategy in RTPs pooling system is delivery and collection methods/rules, which controls inventory via order priority and collection rules. It treats all RTPs in use as the source of inventory. Location strategy determines network layout and service relationships. Depots in RTPs pooling system are versatile. It is responsible for storage, delivery, collection and maintenance at the same time. Due to the reusability of RTPs, the amount of flow within network is fixed. The RTPs in use are new supplements after collection. The total number of RTPs delivered should be equal to the quantity of returned. Transportation strategy helps managers select the means and modes of transportation. Carrier selection and arrangement issues in RTPs pooling system are also worthy of attention. The three strategies are in trade off with each other. That is, a good location strategy is dependent on the manner in which inventories are managed and on the resulting inventory levels, and on the transportation service selected. Inventory levels are dependent on the number and location of facilities as well as the transportation service selected (Ballon, 1997). The triangle of logistics strategy is shown in Figure.1.

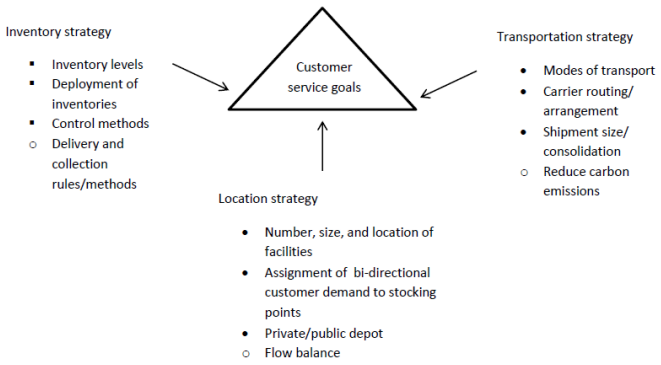


Figure 1: The triangle of logistics strategy

In RTPs pooling system, the demands in forward and reverse market are interrelated. The service levels of two channels are mutually reinforcing. In this situation, the impact of reverse logistics activities on forward service capability is greater than in other CLSC, the issue of making the forward demand and the return flow meet a balanced state is critical for logistics managers. Based on the constraint of traffic balance, the authors propose a service strategy of pooling system from the perspective of resource utilization. It is simulated by software. Resource utilization directly demonstrates the efficiency of logistics operation. As mentioned above, maintaining high service levels, high profitability and market competitiveness through maximizing network resource utilization are the main objectives of RTPs pooling system optimization.

BIDIRECTIONAL SERVICE STRATEGY (BSS)

Various service options are designed to achieve customer satisfaction and maximize the corporate profit. Innovative resource utilization evaluation indexes (RTPs turnover rate, trucks loading rate, collection waiting time etc.) distinguish BSS from the traditional logistics strategy. The BSS contains three sub-strategies: forward flexible service strategy (FFSS), reverse flexible collection strategy (RFCS) and emergency collection strategy (ECS) which will be triggered only when the inventory cannot meet forward customer needs. Literally, the FFSS and RFCS apply to different channels and ECS considers bidirectional needs together. RTPs could serve multiple SC. The average turnover rate of RTPs pooling system is the weighted value of all served SC.

As mentioned above, FFSS focuses on forward channel. It inherits the essence of traditional service strategy, regards the order response speed and order fulfillment rate as service goals. RTPs leasing companies sort the orders based on three service rules and determine order fulfillment rate according to customer acceptable range. Adjustment of the forward service sequence will change the amount and status of reverse demand. All decisions made are to improve resource utilization while providing better service to forward customers.

The forward service capacity and depot RTPs inventory are positively related. When the inventory can't meet customer needs, emergency collection strategy (**ECS**) shall be implemented. ECS will guide the staff to the right places to collect the RTPs, in the right quantity and right manner. If the ECS is triggered, relevant reverse user and depot stock data should be updated and recalculated immediately. Reverse flexible collection strategy (**RFCS**) is an important but easily overlooked subarea in reverse logistics (RL) and closed-loop supply chain (CLSC) researches. Cannella et al. (2016) discuss what factors influence CLSC's performance; indicating that the higher level of return rate increases the robustness of the CLSC. As the forward demand of RTPs pooling is 100% to be returned, RFCS provides two collection methods and economic collection quantity (ECQ) concept aims at identifying appropriate collection pattern for different types of end customers to speed up the return.

For bi-directional operation, the virtual flow of information can be illustrated in the Figure 2. Rectangles represent practical operations guideline and the ovals represent calculation processing and execution criteria. When the orders are received by the logistics department, they will be sorted by some given sorting criteria. Then the comparison between the inventory and the demand according to the sorting sequence ensures that the priority customer is first served. There are two possible results after the comparison: if the inventory meets the needs of all customers, the order will be executed and service provided on demand; otherwise the depot will start an emergency collection program. In reverse channel, returnable quantity and collection date of RTPs that is held by end customers will be calculated when they arrived there. The collection standards are determined based on multiple criteria. The collection task is triggered whenever the amount reaches the ECQ value. Some customers are grouped for collection based on cost and turnover considerations. Emergency collection is not subject to ECQ requirements. It reads the data of returnable RTPs, collects used RTPs back to depot from selected customer and aim to meet customers' needs as soon as possible. In most case, the depot has sufficient capacity to store and repair RTPs; occasionally the returned amount exceeds the maximum storage capacity. The comparison of the depot capacity and customer needs are made sure that the returned RTPs could be cleaned and maintained as soon as possible. After a series of repairs and inspections, those RTPs will be put into the next round of use.

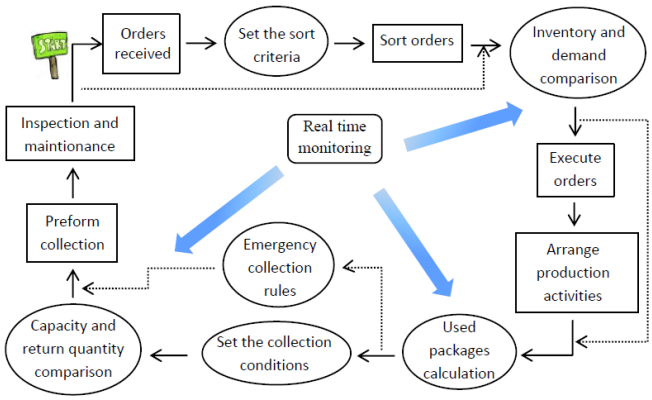


Figure 2: The diagram of data processing loop

Xu et al. (2017) propose a general two-stage quantitative framework that enables decision makers to select the optimal network design scheme for collaborative logistics networks (CLNs) under uncertainty. Phase 1 calculates the simulation result and phase 2 selects the optimal network design scheme for the network design optimization problem under discrete uncertainty (DU-CLNDOP). Learned from this research framework, RTPs bi-directional service strategies for each customer/regional are verified by different scenarios simulation. Step 1 calculates the simulation results using a forward and reverse integration model programming by Python 3.x, and step 2 selects the optimal service strategy for each customer/regional by comparing the key resources utilization indicators.

CRITICAL DATA CALCULATION MODEL

During the operation of RTPs pooling system, the data update and calculations are mainly focusing on tracking the movement of RTPs and determining the execution time of operational tasks. The RTPs held by SC members are in different states: 'need to be maintained' and 'maintained in depot'; safety stock and scheduled in upstream customer; in use and waiting to be collected in end customer. Due to the different production cycle of the supply chains, the daily demand of the RTPs is also dynamically changing.

When the program starts, the model calculates the amount of returnable that day. Comparing the initial value to a set of ECQ standards, if it is greater than any ECQ, the collection task should be performed; otherwise it remains in the waiting state, tagged as 'rest' amount. The returnable amount of next day can be counted as newly leased plus the 'rest' of yesterday. For depot, customers demand is dynamic. If the depot inventory is sufficient to meet customer needs, the orders are executed immediately; if not, the emergency collection will be triggered. Depot collects some used RTPs before they reach ECQ standards, at which time the RTPs returnable amount of these customers becomes 0. The calculation formulas for RTPs holdings are shown in the Figure 3. The parameters used in the calculation are described in detail below.

Quantity variables

In Figure.3, the inventory of the depot q_i^{inv} is equal to original inventory plus all customer returns $\sum_1^K q_{ki}^{ret}$ minus all forward delivery $\sum_1^J q_{ij}^{del}$. Amount of RTPs held by upstream customer q_j^{hel} is equal to the original holding plus the depot delivery q_{ij}^{del} minus the quantity delivered to end customers $\sum_1^K q_{jk}^{tra}$. For end customers, the holding q_k^{hel} is equal to the original holding plus the number delivered from upstream customers $\sum_1^J q_{jk}^{tra}$ minus the number of empty RTPs returned to the depot q_{ki}^{ret}.

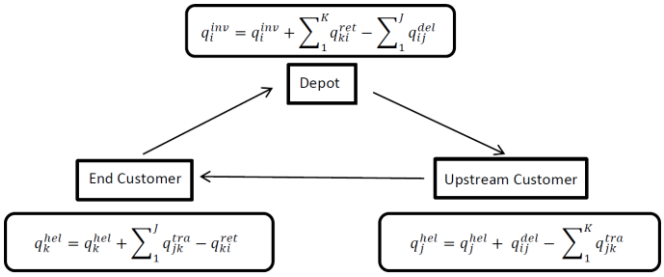


Figure 3: RTPs flow calculation model

Once the upstream customer generates an order, the practical delivery quantity q_{ij}^{del} has three possibilities: when the depot inventory q_i^{inv} is greater than the customer demand q_j^{dem}, then the actual delivery is equal to customer demand q_j^{dem}; when the depot inventory q_i^{inv} is less than the customer demand but greater than the minimum acceptable amount $q_j^{dem} * r_j^{of}$, then the actual delivery is equal to the depot inventory q_i^{inv}; when the depot inventory q_i^{inv} is less than the acceptable minimum order fulfillment rate, emergency collection will be activated. If depot inventory q_i^{inv} plus emergency collection quantity q_i^{ec} is greater than customer demand, then the amount of delivery is q_j^{dem}; if the sum of the two less than customer demand but greater than $q_j^{dem} * r_j^{of}$, then the amount of distribution is original inventory q_i^{inv} plus emergency collection quantity q_i^{ec}.

$$q_{ij}^{del}: \begin{cases} = q_j^{dem}, when\ q_i^{inv} > q_j^{dem} \text{ or } q_i^{inv} + q_i^{ec} > q_j^{dem} \\ = q_i^{inv}, \text{when } q_i^{inv} \in [q_j^{dem} * r_j^{of}, q_j^{dem}) \\ = q_i^{inv} + q_i^{ec}, when\ q_i^{inv} < q_j^{dem} * r_j^{of} \ and \\ \quad q_i^{inv} + q_i^{ec} \in [q_j^{dem} * r_j^{of}, q_j^{dem}) \end{cases} \quad (1)$$

The inventory level is affected by bidirectional demand. When the depot inventory q_i^{inv} is greater than customer demand q_j^{dem}, the remaining amount of inventory is equal to original inventory plus the amount returned minus customer demand. When the inventory/inventory plus emergency collection quantity is between the minimum acceptable amount and the original demand interval, then the RTPs stock will become 0 after the order is performed. Sometimes the amount of inventory plus emergency collection quantity will exceed customer demand, the remaining inventory is equal to original inventory plus emergency collection quantity minus customer demand.

$$q_i^{inv}: \begin{cases} = q_i^{inv} + \sum_1^K q_k^{ret} - \sum_1^J q_j^{dem}, \\ \quad when\ q_i^{inv} + \sum_1^K q_k^{ret} > \sum_1^J q_j^{dem} \\ = 0, when\ q_i^{inv} + \sum_1^K q_k^{ret} \in \left[\sum_1^J q_j^{dem} * r_j^{of}, \sum_1^J q_j^{dem}\right] \ or \\ \quad q_i^{inv} + \sum_1^K q_k^{ret} < \sum_1^J q_j^{dem} * r_j^{of} \ and \\ \quad q_i^{inv} + \sum_1^K q_k^{ret} + q_i^{ec} \in \left[\sum_1^J q_j^{dem} * r_j^{of}, \sum_1^J q_j^{dem}\right] \\ = q_i^{inv} + \sum_1^K q_k^{ret} + q_i^{ec} - \sum_1^J q_j^{dem}, \\ \quad when\ q_i^{inv} + \sum_1^K q_k^{ret} < \sum_1^J q_j^{dem} * r_j^{of} \ and \\ \quad q_i^{inv} + \sum_1^K q_k^{ret} + q_i^{ec} > \sum_1^J q_j^{dem} \end{cases} \quad (2)$$

Formula (3) shows the used empty RTPs may come from one or more end customers.

$$q_i^{ec} = \sum_1^K q_k^{wai} \quad (3)$$

In 3-layers RTPs pooling system, a manufacturer (end customer *k*) can cooperates with multiple suppliers (upstream customer *j*). The total amount of filled RTPs received by end customer *k* is the sum of filled RTPs transferred from different upstream customer *j* on certain day.

$$q_k^{rec} = \sum_1^J q_{jk}^{tra} \quad (4)$$

For the upstream customer, the increase of RTPs comes from the support of the depot *i* and the reduction is caused by the demand of downstream customer *k*. When a supplier (upstream customer *j*) serving multiple manufacturers (end customer *k*), the remaining amount of the empty RTPs held by the supplier *j* can be calculated by original holdings q_j^{hel} plus the number of empty RTPs delivered by depot *i* q_{ij}^{del} minus the sum of filled RTPs transferred to downstream customers $\sum_1^K q_{jk}^{tra}$.

$$q_j^{hel} = q_j^{hel} + q_{ij}^{del} - \sum_1^K q_{jk}^{tra} \quad (5)$$

In reverse channel, if the sum of used empty RTPs waiting to be collected is less than the economic collection quantity (ECQ), then the amount of returned RTPs is 0, the total wait becomes the old wait q_k^{wai} plus new released q_k^{col}; otherwise the returned amount is equal *nECQ* and the remaining amount is the sum of used RTPs minus *nECQ*. When the emergency collection is activated, if the sum of collected plus the original inventory is greater than the minimum demand of customer, the amount returned is equal to the number of RTPs waiting to be collected that is held by the selected customer.

$$q_{ki}^{ret}: \begin{cases} = 0, & when\ \ q_k^{wai} + q_k^{col} < ECQ \\ = nECQ, & when\ q_k^{wai} + q_k^{col} \geq ECQ, \\ = q_k^{wai},\ when\ q_i^{inv} + \sum_1^K q_k^{ret} < \sum_1^J q_j^{dem} * r_j^{of}, \\ \quad q_i^{ec} = \sum_1^K q_k^{wai}\ and\ q_i^{inv} + \sum_1^K q_k^{ret} + q_i^{ec} \geqq \sum_1^J q_j^{dem} * r_j^{of} \end{cases}$$

(6)

The quantity of empty and filled RTPs q_k^{hel} held by end customer k can be calculated by filled RTPs in production plus empty RTPs in waiting or original holdings plus total receiving minus total return.

$$q_k^{hel} = q_k^{wai} + q_k^{pro} \quad (7)$$

$$q_k^{hel} = q_k^{hel} + \sum_1^K q_{jk}^{tra} - q_{ki}^{ret} \quad (8)$$

Assuming in a multi-layer network, if the RTPs could be reused without maintenance and only the customers who locate in the first layer are the RTPs input ports, the parameters and flow balance constraints among customers at different layers are shown as below:

$$\sum_{T+PC} \sum_I \sum_{EC} q_{ec}^{ret} = \sum_T \sum_I \sum_{FC} q_{fc}^{rec} - \sum_C q_c^{hel} \quad (9)$$

$$q_{fc}^{del} = q_{fc}^{rec} - q_{fc}^{hel} \quad (10)$$

$$q_{c_u}^{del} = q_{c_u}^{rec} - q_{c_u}^{hel} \quad (11)$$

$$q_{ec}^{ret} = q_{ec}^{rec} - q_{ec}^{hel} \quad (12)$$

Constraint (9) clarifies the relationship between forward reception and reverse collection. By the end of the planning period, the RTPs returned from end customer q_{ec}^{ret} equal to the total amount received by first layer customer q_{fc}^{rec} minus customer holdings q_c^{hel} at that time. The 'c' refers to customers at all layers. Constraints (10)-(12) indicate the throughput of RTPs at all layers. The delivery/return amount from the previous layer to the next layer is equal to their received amount minus the holding amount.

In the previous situations, the RTPs receipts and returns occur at specific customer layer. The role of each participant is fixed. When all customers can order RTPs directly from the depot, customers belong to different channels at different stages. Customers could receive loaded RTPs from upstream customer and order empty RTPs from the depot simultaneously. The used empty RTPs could be returned to nearby depot from customers when there is no more demand. The boundaries between forward and reverse customers in the system become blurred, while the flow simulation between customers at different layers and depots becomes more complicated.

$$\sum_T \sum_I \sum_C q_{ic}^{del} = \sum_T \sum_I \sum_C q_{ci}^{ret} + \sum_C q_c^{hel} \quad (13)$$

$$q_c^{rec} = q_{c_{uc}}^{del} + q_{ic}^{del} \quad (14)$$

$$q_c^{ret} = q_c^{rec} - q_{cc_d}^{tra} - q_c^{hel} \quad (15)$$

When any member of supply chain could place an order with the RTPs depot, there are no strict tier restrictions in the system. Constraint (13) makes sure that the total delivery from depot q_{ic}^{del} is equal to the total returned from customer q_{ci}^{ret} plus their holdings q_c^{hel} by the end of planning period. Constraint (14) ensures the source of the RTPs is upstream customer c_u and depot i. Constraint (15) clarifies that the amount of RTPs received q_c^{rec} minus the amount of RTPs transferred $q_{cc_d}^{tra}$ between customers c and c_d and minus the holding quantity q_c^{hel} is equal to the amount returned q_c^{ret}.

Time variables

Once the depot i delivers the empty RTPs to upstream customer j, the leasing contract takes effect. The date the upstream customer j receives the empty RTPs from depot i is from the departure date plus the number of t_{ij} transportation days.

$$t_j^{rec} = t_{ij}^{del} + t_{ij} \quad (16)$$

It takes t_j^{fil} days for the empty RTPs to be filled with different raw materials after they arrive at upstream customer j. Fully loaded RTPs are sent to the designated end customer after they are filled.

$$t_{jk}^{del} = r_j^{rec} + t_j^{fil} \quad (17)$$

End customer k will receive the fully loaded RTPs after t_{jk} days of shipping.

$$t_k^{rec} = t_k^{del} + t_{jk} \quad (18)$$

Assume the production cycle of manufacturer (end customer k) is stable during planning period, after a production cycle t_k^{pc}, the used empty RTPs could be collected on a certain date t_k^{col}.

$$t_k^{col} = t_k^{rec} + t_k^{pc} \quad (19)$$

Considering the cost issue, RTPs collection needs to reach an economic collection baseline. The actual return may take some days t_k^{wai} to reach the standard. As the waiting time t_k^{wai} is affected by the amount of used RTPs, logistics managers could shorten the length of this phase by adjusting or developing new operation methods.

$$t_k^{ret} = t_k^{col} + t_k^{wai} \quad (20)$$

When the amount of used RTPs held by the end customer k meets the collection requirements, the depot i will receive the RTPs on a date after the number of t_{ki} transportation days.

$$t_i^{rec} = t_{ki}^{del} + t_{ki} \quad (21)$$

In the bulk or chemical cargo supply chain, in order to ensure that raw materials are not contaminated by packaging, each package must be cleaned and maintained before they are ready for service again. The day the empty RTPs could serve the customers should be after a maintenance period t_i^{mai}.

$$t_{ij}^{del} = t_i^{rec} + t_i^{mai} \quad (22)$$

Equation (23) calculates the length of time W_k for the used RTPs to wait for collection which is held by customer k. The returnable amount of RTPs in each batch is different. When the current returnable quantity is insufficient to reach the ECQ, the manufacturer will continue to hold the used empty RTPs until the sum of the quantities can meet the collection requirements. The waiting time for the same batch of collected RTPs may also be different. Waiting time for each cycle can be calculated by multiplying the collectable quantity by the time different between RTPs actual return and can be collected.

$$min\, W_k = \sum_T \sum_K q_{ki}^{col} * (t_{ki}^{ret} - t_k^{col}) \quad (23)$$

CASE STUDY

A foldable RTP leasing company was selected as a simulation case. The data of two depots (i), 15 suppliers (j) and 63 manufacturers (k) were collected from January to August 2013. Suppliers and manufacturers are sorted according to the amount of demand.

Table 1 lists the delivery frequency between depot i and supplier j. Customers are ranked in descending order of annual demand The delivery frequencies of the top 3 suppliers are far more than that of others. In forward channels, the large customer has services priority in fixed customer priority and is uncertain when sorted by daily order size. Manufacturers working with them will also receive more efficient services than in other supply chains. The frequency of forward transportation shows the size of customer demand and determines the importance of the customer.

Table 1: The frequency of forward transportation

J_n	fre_{ij}	J_n	fre_{ij}	J_n	fre_{ij}	J_n	fre_{ij}
J_1	67	J_5	11	J_9	1	J_{13}	3
J_2	64	J_6	16	J_{10}	2	J_{14}	0
J_3	40	J_7	5	J_{11}	2	J_{15}	3
J_4	18	J_8	2	J_{12}	1		

Figure 4 indicates the collection frequency of reverse channel. In the same area, the number of manufacturers (63) is almost 4 times greater than the supplier (15). 3 manufacturers performed collection more than 20 times; 9 manufacturers' collection between 16 and 10 times; another 12 range from 9 to 5 times, the rest are between 4 to 0 times. Most manufacturers collected 3 times to 16 times. The number of total collection is far beyond the number of delivery. This demonstrates the necessity of reverse system improvement from an economic perspective.

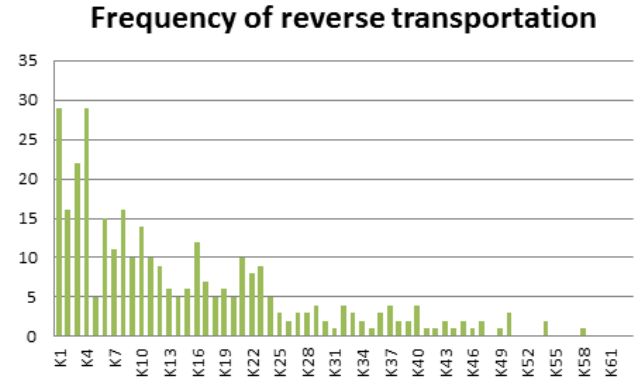


Figure 4: The frequency of reverse transportation

After a production cycle, the used empty RTPs could be returned to nearby depot when the quantity meets the collection requirements. The waiting time equals invalid idle time, shortening it means speeding up the turnover of the RTPs. Figure 5 shows the total waiting time of RTPs. The curve is gradually reduced. Some used RTPs are collected immediately as soon as they finish tasks. The waiting time for these RTPs is 0. Taking into account the collection quantities of the RTPs, the curve of average waiting time for each manufacturer is opposite to the total waiting time shown in the Figure 5. Different scenarios can be simulated to choose the appropriate strategy for supply chain members. The waiting time filters out inefficient points then points out the area that may be optimized.

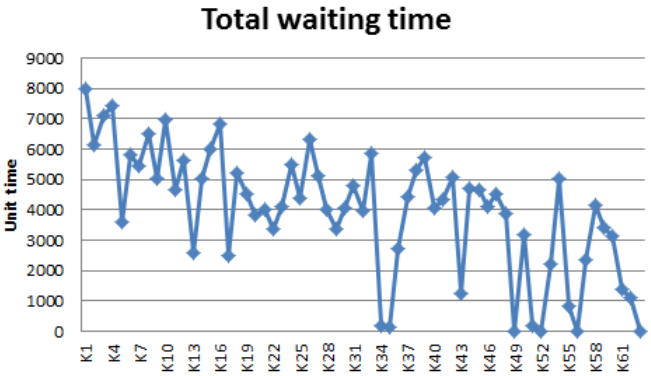


Figure 5: Total waiting time of manufacturer (k)

Table 2 lists the date and the actual amount of emergency collection. They mainly happened in March and April. According to these data, logistics managers could deploy more resources to this area during this time period to cope with the spike in demand.

Table 2: Emergency collection data and amount

Date	Amount	Date	Amount	Date	Amount
17-03	96	01-04	128	08-04	92

19-03	133	02-04	128	09-04	147
27-03	52	03-04	162	10-04	98
28-03	69	04-04	105	12-04	144
29-03	83	05-04	103	13-04	113
31-03	90	07-04	104		

Figure.6 and 7 illustrate the inventory fluctuations during the planning period. As shown in Figure 6, the inventory in January fell rapidly and rose sharply in February. It is caused by the return of used RTPs after a production cycle. Consistent with market demand growth, inventory levels fell back to the bottom in March and April. The curve fluctuates steadily within 2500-3500 from May to June. Most used RTPs were returned in July. After May, about 1,500 RTPs are underutilized and they can be deployed to other regions to meet market demand

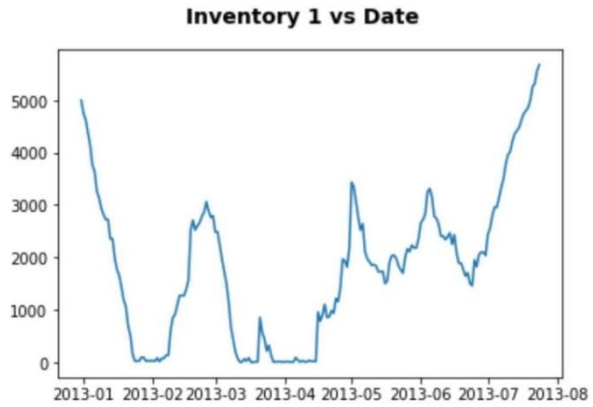


Figure 6: The inventory fluctuations of depot 1

In Figure 6, the total amount of inventory at the end of July is more than the initial amount and in Figure 7 is less than. Some RTPs were moved from depot 2 to depot 1.

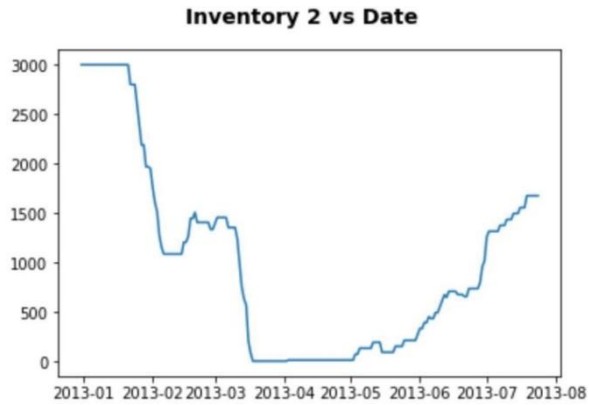


Figure 7: The inventory fluctuations of depot 2

The curve in Figure 7 shows differences compared to Figure 6. As an alternative depot to some customers, Depot 2 provides services when the inventory of the preferred depot does not meet the demand. The curve continues to decline until May. Depot 2 wills no longer support when the depot 1 is fully stocked. The curve rose steadily after May.

The fluctuation of two depot curve is similar to the depot 1. Total inventory at the end of planning period is less than initial input, it indicates that some used RTPs still await at manufacturers and some suppliers holds some RTPs as safety stock.

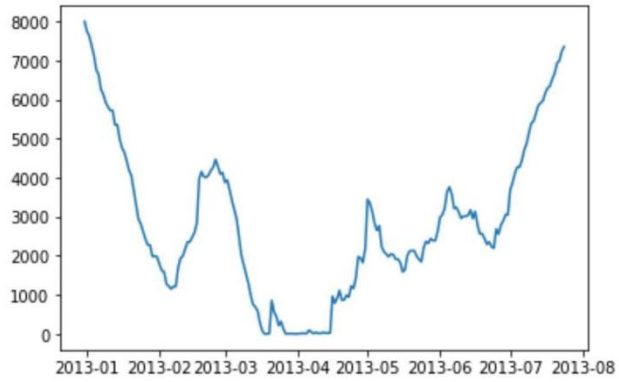


Figure 8: The sum of inventory

CONCLUSION

In RTPs pooling system, in addition to general research targets (service efficiency and total cost), the resource utilization deserves more attention. From an operational perspective, reverse logistics is as important as the forward. Both of them impact vehicle arrangement and RTPs turnover rates. From a structural point of view, the number of supply chain layers and flow directions are critical in determining the service criteria between depots and customers.

This paper proposes a bidirectional service strategy for RTPs pooling system. Orders are sorted according to customer priority; trucks are scheduled based on the size of demand; collection plan is designed to be flexible; inventory level directly reflects the operational efficiency, which is controlled by both forward and reverse strategies; emergency collection is triggered when the inventory is insufficient; and the production cycle is given by different supply chains. Due to the space limitation, this article mainly introduces the service strategy and explains the tracking and function of key data in strategy execution.

The frequency of forward transportation shows the size of suppliers' needs and the importance of the supplier. The collection frequency filters out inefficient points then shows the area that may be optimized. The waiting time equals the invalid idle time, shortening of which means speeding up the turnover of the RTPs. Different scenarios can be simulated to choose the appropriate strategy for customers. According to emergency collection data, logistics managers could deploy more resources to this area during this time period to cope with the rise in demand. Inventory curve indicates the change of market demand and could be used as a numerical basis for resource deploy. All of them can be compared underlying different operation options. In conclusion, an appropriate operation schedule can shorten invalid collection

waiting time and keep regional flow balance. Compare with traditional logistics strategy, bidirectional service strategy regards forward and reverse demand as 'liner correlation'. Forward order priority has positive impact on reverse collection needs. Adjustments to the collection methods will also affect the inventory level. It helps logistics managers to identify core areas and adjust their logistics strategies via:

1) Tracking inefficient customer;
2) Tracking seasonal demand fluctuations;
3) Tracking bidirectional traffic in the area.

In order to promote the sustainable development of the logistics industry, the future research could pay more attention to the application of reusable articles and simulation method of multi-layer service network. More operation options could be added into service strategy as appropriate to the actual conditions.

REFERENCES

Ballou, Ronald H., Business logistics — importance and some research opportunities, Scientific electronic library online, Gest. Prod. vol.4 no.2 São Carlos, 1997, http://dx.doi.org/10.1590/S0104-530X1997000200001

Bolboaca, Codruta., COOPERATION STRATEGIES FOR LOGISTICS SERVICE PROVIDERS IN THE ART TRANSPORTATION MARKET, Managerial Challenges of the Contemporary Society. Proceedings, 2010, 20-25

Cannella, Salvatore., Bruccoleri, Manfredi., Framinan, Jose M., Closed-loop supply chains: What reverse logistics factors influence performance? International Journal of Production Economics, 2016, Vol. 175, pp 35-49

Cooper, Martha C. and Ellram, Lisa M., Characteristics of Supply Chain Management and the Implications for Purchasing and Logistics Strategy, The International Journal of Logistics Management, 1993, Vol. 4, pp.13-24

Ellram, Lisa M. and Cooper, Martha C., Supply chain management, partnerships, and the shipper-third party relationship, International Jounal of Logistics Management, 1990, Vol.1, pp 1-10

Gallego, Carrasco., Cueto, Ruth and Ponce., María, Eva., A management model for closed-loop supply chains of reusable articles: defining the issues. 2010, In: "4th International Conference On Industrial Engineering and Industrial Management/XIV Congreso Ingeniería de Organización", 08/09/2010 - 10/09/2010, San Sebastian, Cantabria. ISBN ISSN 1132-175X

Hertz, Susanne., Alfredsson, Monica., Strategic development of third party logistics providers, Industrial Marketing Management, 2003, Vol.32, 139-149

Kroon, L; Vrigens, G. Returnable containers: an example of reverse logistics. International Journal of Physical Distribution and Logistics Management. 1994, Vol 25. 56-68

Lee, Kuoliang., Two-phase fuzzy approach for evaluating service strategies in an airport's international logistic system, Journal of the Chinese Institute of Industrial Engineers, 2009, Vol.27, 209-225

Liu, Weihua., Liu, Xiaoyan., Li, Xiang., The two-stage batch ordering strategy of logistics service capacity with demand update, Transportation Research Part E: Logistics and Transportation Review, 2015, Vol.83, 65-89

McGinnis, M.A. and Kohn, J.W., Logistics strategy-revisited, Journal of business logistics, 2002, Vol.23,1-18

McKerrow, Don., What makes reusable packaging systems work, 1996, 96 Conference, 19-20 March 1996, Brussels, Belgium, organized by Ryder Worldwide Conferences Ltd.

Mensendiek, Arne., Scheduling with returnable containers, Journal of Scheduling, 2015, Vol 18, Issue 6, pp 593–605

Porter, M.E., What is strategy? Harvard Business Review, 1996.

Rudolf, Leuschner. and Douglas M, Lambert., Establishing Logistics Service Strategies That Increase Sales, Joutnal of business logistics, 2016, Vol.37, 247-270

Sarkar,Biswajit., Ullah,Mehran., Kim,Namhun., Environmental and economic assessment of closed-loop supply chain with remanufacturing and returnable transport items, Computers & Industrial Engineering, Volume 111, 2017, pp 148-163

Soysal, Mthmet., Closed-loop Inventory Routing Problem for Returnable Transport Items, Transportation Research Part D, 2016, Vol.48, pp 31-45

Twede, D and Clarke, R., Supply chain issues in reusable packaging, Journal of Marketing Channels, 2005, Vol.12(1), 7-26

Zhang, D. and Clausen, U., Time-oridnted collection schedule simulation in closed-loop RTP network with multiple economic collection quantities and multiple production cycle, The 2016 EUROPEAN SIMULATION AND MODELLING CONFERENCE, 440-447

Xu, X.F., Hao, J., Deng, Y.R., Wang, Y., Design optimization of resource combination for collaborative logistics network under uncertainty, Applied Soft Computing, 2017, Vol. 56, 684-691.

Martin Murray, 2017, https://www.thebalance.com/creating-a-logistics-strategy-supply-chain-management-2221277

Web reference 1, http://lexicon.ft.com/Term?term=logistics-strategy

AUTHOR BIOGRAPHIES

Di Zhang studied logistics network simulation and optimization at the TU Dortmund, Germany. Since 2012 she is PhD student in the institute of transport logistics. Her main research interests are sustainable logistics networks and their application for practical industries.

Uwe Clausen is the managing director of the Institute of Transport Logistics at the TU Dortmund University and director of the Fraunhofer-Institute for Material Flow and Logistics (IML). He is amongst others chairman of EffizienzCluster LogistikRuhr and a member of the Scientific Advisory Board of German Logistics Association (BVL).

INTEGRATED ELECTRIC VEHICLE SCHEDULING AND CHARGING INFRASTRUCTURE PLANNING FOR PUBLIC TRANSPORT

Hubert Büchter
Fraunhofer-Institute for Material Flow and Logistics IML
Joseph-von-Fraunhofer-Str. 2-4
44227 Dortmund
Germany
E-mail: hubert.buechter@iml.fraunhofer.de

Sebastian Naumann
ifak - Institut f. Automation und Kommunikation e.V.
Werner-Heisenberg-Str. 1
39106 Magdeburg
Germany
E-mail: sebastian.naumann@ifak.eu

KEYWORDS

Optimization, Simulation, Transportation, Urban affairs, Vehicle Scheduling

ABSTRACT

Electric buses are increasingly being used in local public transport. Although large capacity batteries are available, there must be at least one charging station. Buses with smaller batteries in particular require a well thought-out charging infrastructure. Planning an optimized charging infrastructure is a complex and time-consuming optimization process. In the end, no bus may give up due to a lack of power. Although planning is a non-real time problem, a short runtime is desirable.
A charging infrastructure is a precondition for planning the bus schedule. The possible sequences of trips depend not only on times but as well on the state of charge of the batteries. There are mutual influences between the bus schedule and the charging infrastructure. The prerequisite for planning the infrastructure is a valid bus schedule and vice versa. Once a feasible plan has been found, these steps of planning can be executed iteratively.
The infrastructure is optimized by a simulation driven Branch and Bound algorithm with a specific heuristic. For planning the bus schedule, Column Generation is used. All algorithms are part of a planning tool for bus companies, energy providers and urban planners.

INTRODUCTION

The global trend towards clean and energy-efficient vehicles is driven by concerns regarding the impacts of fossil-fuel-based road transport on energy security, climate change and public health. In particular, the electrification of road transport is seen as one of the many possible ways to use energy from renewable sources. This leads to a reduction of local emissions and greenhouse gas emissions.
The requirements of electric busses in public transportation are different to those of electric passenger cars. A bus covers an average distance of 250 to 300 km each day. The bus itself usually has a weight of approximately 10-25 t. A suitable battery that would enable the bus to run for such a long distance without having to be recharged would be far too big, heavy and expensive. Bigger batteries with an overnight charging or medium sized batteries with a rapid charging concept might produce unwanted power peaks. A charging infrastructure, which supports the power grid, requires an appropriate amount of charging stations with the aim to spread the load in space and in time.
Running a fleet of electric buses requires very high investments both in vehicles and in charging stations. Sophisticated planning algorithms are required in order to keep the investment costs as well as the operation costs as low as possible. We show how the vehicle scheduling and the charging infrastructure planning can exploit their strong mutual dependency to achieve this aim.

This paper focuses on public transport with electrically driven buses and charging their batteries at already existing bus stops. It also includes solving the problem of vehicle scheduling taking the state of charge of the batteries into account. The tasks are:

- Selecting those bus stops, which must become charging stations,
- minimizing the number of required buses and
- building sequences from all trips and assigning them to the buses.

The currently implemented planning process minimizes investment and operation costs for buses and for charging devices. An extension to other e.g. technical aspects such as minimum overlapping time for charging different buses is possible.

This paper is structured as follows: We start with a look at related work. The following section introduces the problem and describes the objects and their interaction and properties. The next section introduces the bus schedule problem with respect to the recharging problem. Two sections describe the simulation and optimization models for the infrastructure planning. Then the closed loop for both planning algorithms follows. Finally, we present some results and give an outlook for the further work. This work is part of the project 'MENDEL' (Trumpold et al. 2017) that is funded by the German government.

RELATED WORK

The problems of vehicle disposition and charging infrastructure planning are analyzed separately in many papers. Sassi and Oulamara (2014a; 2014b) solve the problem of *vehicle scheduling and optimal charging* on a given charging infrastructure. The fleet consists of electric and combustion vehicles. They provide a mixed-integer linear programming formulation to model the problem and solve small and medium instances.

Kooten Niekerk et al. (2017) introduce a scheduling algorithm for electric vehicles running according to a time table. They also assume a given charging infrastructure.

Many publications about the planning problem for charging infrastructures deal with non-public urban transport, e.g. (Frade et al. 2011; Kuchshaus et al. 2012; Chen et al. 2013). An overview of optimal planning for charging stations is in (Zheng et al. 2014). That work deals with stochastic models with the focus on the grid and does not meet the specific characteristic of public transportation, which is timetable driven. The complexity of the planning process for charging infrastructure is analyzed in (Lam et al. 2014) and it shows that it is NP complete. Due to this result, any planning algorithm of practical relevance is an approximation. Olsen and Kliewer (2016) extend a vehicle-scheduling problem by charging models with different battery characteristics. They find that in a very often used operation range a linear approximation of the battery characteristic is sufficient. Buechter and Naumann (2016a; 2016b) use this approach in an optimizer that is based on linear programming and simulation. This optimizer solves the problem for linear behavior of all components but with a huge demand of memory. It is limited to about 100 buses and 1000 bus stops, which is too less for many applications.
Espinouse et al. (2000) introduce a simulator driven optimizer based on branch and bound, which is very specific for job sequencing and vehicle dispatching. In our problem, there is no need for sequencing but many potential solutions are not feasible.

PROBLEM DESCRIPTION

The problem consists of two parts: The bus schedule problem, which is also called Vehicle Schedule Problem (V*SP*) and the infrastructure planning (*ISP*). Both planning procedures work on a *system model*. The relevant physical objects, which the system model takes into consideration, are *buses*, *ways*, *bus stops*, *charging stations* and *trips* (1).

$$\left.\begin{array}{rlll} \text{Buses} & i \in \mathbb{B}, & \mathbb{B} & = \{1, \dots, nbBuses\} \\ \text{Ways} & j \in \mathbb{W}, & \mathbb{W} & = \{1, \dots, nbWays\} \\ \text{Bus Stops} & k \in \mathbb{S}, & \mathbb{S} & = \{1, \dots, nbStops\} \\ \text{Charging Stations} & & \mathbb{C} & \subseteq \mathbb{S} \\ \text{Trips} & l \in \mathbb{T} & \mathbb{T} & = \{1, \dots, nbTrips\} \end{array}\right\} \quad (1)$$

The transportation network consists of bus stops $\mathbb{S}$ and ways $\mathbb{W}$. A way is a sequence of roads connecting two bus stops. The corresponding mathematical model is a directed graph $N = (\mathbb{S}, \mathbb{W})$. The nodes $\mathbb{S}$ represent the bus stops and the edges $\mathbb{W}$ represent the ways each connecting two bus stops. Each way is attributed with length l and average values for drive time t_D and energy consumption e^-.

Buses can charge at those bus stops on their routes, which are equipped with a charging device. There are some limits for available power, power transfer and battery energy (Table 1).

Table 1: Power and Energy Limits for Buses and Charging Stations

Bus		Charging Station	
$\boldsymbol{pb_{max}}$	Max. charging power	$\boldsymbol{ps_{max}}$	Max. power offer
$\boldsymbol{p_{base}}$	Base power consumption		
$\boldsymbol{e_{min}}$	Min. battery energy		
$\boldsymbol{e_{max}}$	Max. battery energy		

Batteries have limited charging power. This limit can be individual for each bus ($\boldsymbol{pb_{max}}$). The charging devices have a maximal output power. This limit can be individual for each charging station ($\boldsymbol{ps_{max}}$).
If a bus stop is equipped with a charging device, it becomes a *charging station* $\mathbb{C} \subseteq \mathbb{S}$. The estimation of $\mathbb{C}$ is part of the optimization model, which is introduced in the following section. The system model also describes the charging behavior of the batteries depending on specific characteristics.
Figure 1 shows a typical energy and power flow.

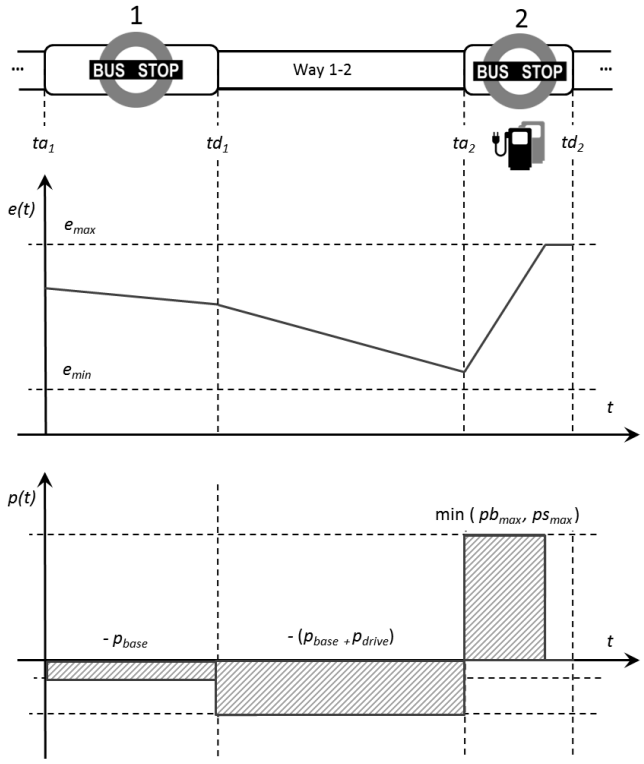


Figure 1: Energy and Power Flow for One Bus Arriving at Bus Stop 1 at t_{a_1} Starting at t_{d_1} and Driving over Way 1-2 to Charging Station at Bus Stop 2

The buses' charging controllers may make full use of the energy offer or even in part. Less charging power is preferable to reduce power peaks on the grid and to save battery life.

BUS SCHEDULING

The main input data to the vehicle scheduling in general are the trips of the timetable, which have been defined on previous stages. On a trip of the timetable, the responsive electric vehicle transports passengers on a fix chain of stops, where the passengers can enter and leave the vehicle. In a more formal way, a trip therefore is defined by a list of stops s_1, s_2,

…, s_n and arrival times $t_{a,i}$ (arrival time at stop i) and departure times $t_{d,i}$ (departure time at stop i).
Electric vehicle scheduling is the assignment of the trips from the timetable to tasks. A task is the schedule of a vehicle for a certain day, including the pull-out from the depot, a sequence of trips from the timetable, any dead-head trips between the trips from the timetable, charging operations and the pull-in back to the depot. Whereas the time and mileage of the trips are determined by the timetable, vehicle scheduling aims to minimize the unproductive periods of time (e.g. pull-ins, pull-outs, dead-head trips, charging operations) and the number of required vehicles with the following constraints:

- Each trip of the timetable must be covered by a vehicle.
- A vehicle cannot be assigned to more than one trip simultaneously.

Each task must be feasible for the electric vehicle such that the state of charge of the vehicle's energy storage never runs below a defined threshold.

The aim of the vehicle scheduling is to find a minimum number of routes from the depot start node to the depot terminal node, where each node is visited exactly once and no battery energy falls below a lower bound. For each route, one bus is used and so the number of required buses is minimal.
The structure of the result is a schedule like Table 2.

Table 2: Extract from a Sample Bus Schedule

Bus Number	Departure Time	Way Number	Arrival Time	Bus Stop Number
…	…	…	…	…
1	08:20	2	08:23	540
2	08:20	798	09:08	779
1	08:25	4	08:52	544
3	08:30	2	08:33	540
1	08:55	7	09:05	555
2	09:12	943	10:10	622
…	…	…	…	…

This kind of schedule is the input for the infrastructure planning.

We implemented the method of electric vehicle scheduling described in (Kooten Niekerk et al. 2017). Here, linear optimization together with a column generation approach is applied. We created a network where each node represents a trip and where the arcs between nodes represent the compatibility of the each connected trips. Compatibility means, that the two connected trips could be solved by one vehicle considering arrival and departure times as well as transition times. (Kooten Niekerk et al. 2017) introduces additional nodes for charging locations. Trip nodes as well as charge nodes are split into separate nodes for each state of charge (SOC) value (see Figure 2). This model is called 'discrete charge model'. In order to achieve high quality results, the SOC interval should be chosen as small as possible, however, the high number of nodes and arcs might quickly exceed memory and CPU resources.

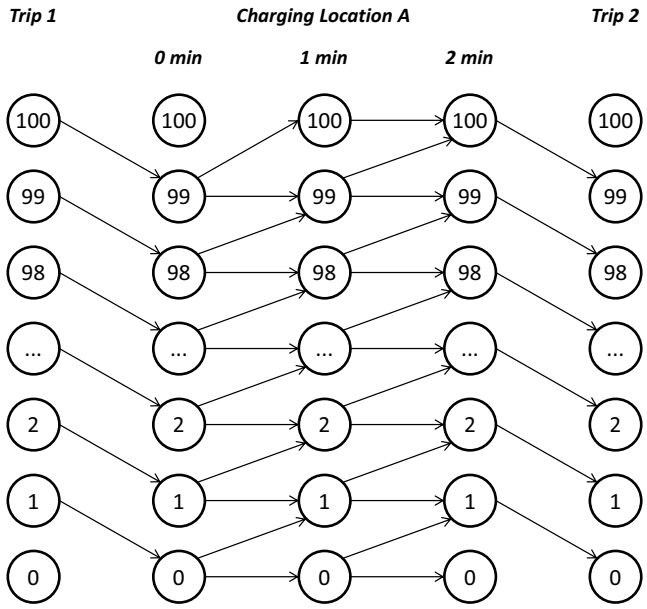


Figure 2: Discrete Charge Model: Sample for two trips and a charging location

When generating new columns (i.e. tasks), the subproblem to be solved is to find the shortest path within the discrete charge model from the start depot to the end depot whereas the arcs have negative costs. The result of the linear optimization step is the set of the generated columns (i.e. tasks) with the least total costs while covering all trips and therefore a valid schedule for the electric vehicles including charging operations.

INFRASTRUCTURE PLANNING

Infrastructure planning calculates the demand for chargers at bus stops. Charging may be prohibited at some bus stops and some others may already have chargers. The last point arises when the changeover from Diesel buses to electric buses is made in stages.
The planning is based on a Branch-And-Bound Optimizer, which minimizes investments for all chargers. The optimizer selects subsets of bus stops as candidates for upgrading to charging stations. In addition to the economic aspect, the solution must also be feasible so that all buses can perform their services without energy lack. This restrictive constraint can be checked by simulating the entire bus schedule.
In order to select the subsets more effectively, it is better to estimate a degree of feasibility than to consider only a binary result for this technical constraint.
The next two subsections explain the simulator and the optimizer in detail.

Simulation Model

The simulation model is deterministic and works on a set of charging stations $\mathbb{C}$ and the maximum power $\boldsymbol{ps_{max}}$, which they can deliver. At the beginning of simulation, all buses are placed at their initial bus stop, the battery energy is set to

the initial value e_{ini} and the simulation clock is set to the start time t_0.
The simulator runs event driven. There are two types of events: Departure from a bus stop and arrival at a bus stop. When an event occurs, the charging states of the batteries are updated and some data is collected for later evaluation. The energy of the batteries must always be in a range between e_{min} and e_{max}.
There are two different simulation modes: *Full simulation* and *embedded simulation*. Full simulation collects many data while running and it does not stop until all trips will have terminated. If the battery's charge of one or more buses falls below e_{min} and even if one or more batteries are temporarily in a negative charging state, which is physically impossible, the simulation process continues. The purpose is to find bottlenecks manually e.g. by analyzing some plots of battery charge versus time or battery charge versus number of charging stations visually.
The embedded simulation acts strictly. If at least one bus's energy falls below e_{min} then the simulation stops and returns a degree of progress. This degree in a range $[0,1]$ is a measure of the feasibility of the set of charging stations. The optimizer uses this mode internally thus reducing the computation time significantly.
When a bus arrives at a bus stop the energy consumption for the drive from the previous bus stop is to calculate and to subtract from the current bus's energy e. For a more sophisticated model it is not only a subtraction but it also follows a specific nonlinear battery characteristic. The precision of the model is up to the requirements. The energy demand depends on the type of the bus, the driven way and the drive time for this way. The consumed base energy is given by $p_{base} \cdot t_{drive}$.
When a bus leaves a bus stop, the battery energy is reduced by the base energy which is $p_{base} \cdot t_{stop}$. If the bus stop is a charging station then the calculation of the power flow is a bit more complicated. Arriving at those bus stops the charging process starts if the bus's battery energy has fallen below a given value and if the stop time is greater than the set-up time for connection plus the set-down time for disconnection. During the stay at a charging station the charging device delivers the base power and the charging power. The transfer power $p_{transfer}$ is the minimum of the power offer p_{stop} from the charger and the maximum allowed charging power p_{max} of the battery and the base power p_{base} (2).

$$\left.\begin{aligned} p_{transfer} &= p_{charge} + p_{base} \\ p_{transfer} &= \min(p_{stop},\, p_{max} + p_{base}) \\ p_{charge} &= \min(p_{stop} - p_{base},\, p_{max}) \end{aligned}\right\} \quad (2)$$

The power flow p_{charge} for charging can be negative if the charger does not deliver enough power to satisfy the base power demand.

Simulation Based Optimizer

The aim of the optimizer is to estimate from a given set Y of bus stops a subset X of charging stations which minimizes the value of the *objective function* $c(X)$ under the constraint $f(X) = 1$. See (3) for the problem definition.
Let the function $f(X)$ express a value for the *feasibility* of the set X. The solution is feasible if $f(X) = 1$. Otherwise, $f(X)$ is a measure for the benefit of X. The better the feasibility the more suitable is the vector X.
The condition for the existence of a solution is the feasibility of the origin set (4)

$$X \subseteq Y \quad \begin{cases} c(X) \to \min & c \in [0,\infty] \\ f(X) = 1 & f \in [0,1] \end{cases} \quad (3)$$

$$f(Y) = 1 \quad (4)$$

$$U \subseteq V \subseteq Y \quad \begin{cases} c(U) \le c(V) \\ f(U) \le f(V) \end{cases} \quad (5)$$

The objective function reflects very often the economic costs. However, other aspects like urban planning factors, risks or technical efforts could be considered too. Both, the objective function $c(X)$ and the simulator function $f(X)$ are monotonic (5) which is given in the context of the planning problem. If more charging devices are installed then the costs increase but also the feasibility of the system increases.

There are some specific requirements for the infrastructure planning process. Some charging stations may already exist and some bus stops cannot be equipped with charging devices. A second requirement is a short run time for the optimizing process. The problem is NP-hard and without any heuristics, the run time of any implementation would be unacceptable. The following paragraphs present a solution for these problem specific requirements.
A simple solution is to set the costs for existing charging stations E to zero and those for the excluded bus stops F to infinity. However, in order to reduce the solution space it is better to give the set E and the set F in advance. These sets are defined in (6) and (7) shows the influence to the resulting set X.

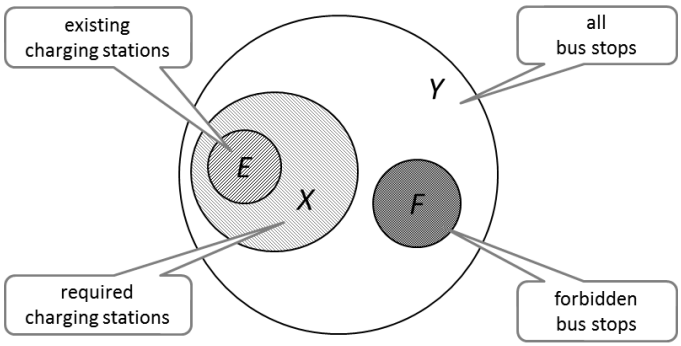


Figure 3: Dependencies between Existing Bus Stops, Forbidden Bus Stops, Existing Charging Stations and the Solution X

$$F \subseteq \mathrm{Y}, \quad E \subseteq \mathrm{Y}, \quad F \cap E = \emptyset \quad (6)$$
$$X \subseteq (\mathrm{Y} \backslash \mathrm{F}) \cup E \quad (7)$$

The reduction of the problem size has a significant influence on the run time. It is applicable to many real bus systems because many bus stops are not suitable to become a charging station.
The optimizer is based on the Branch and Bound algorithm (Algorithm 1). The lower bound is given by an objective function for invest for feasible sets of charging stations, which are selected. Real world problems cannot be optimized in an acceptable run time without any heuristic.

It depends on the access strategy of the queue how the solution set develops. Three basic branching strategies are very common:

- *First In First Out*: The search tree is developed in the breadth first (BFS). The implementation of the queue has the semantic of a sequential queue (FIFO).
- *Last In First Out*: The search tree is developed in the depth first (DFS). The implementation of the queue has the semantic of a stack (LIFO). This strategy is often implemented by recursion, which uses implicitly the call stack of the runtime environment.
- *First In Best Out*: The search tree follows always the best candidates first (BestFS). The implementation of the queue has the semantic of a priority queue.

The optimizer for the planning process uses a combination of all three basic strategies. It starts with a breadth search and puts the results in a priority queue with a limited capacity k (Algorithm 1, line 20). This reduces the search space significantly but with the risk for losing the solution for the global optimum. The priority is calculated as the relation between the feasibility and the costs of the current selected subset. This is a measurement for the use costs relation, which finally represents the selection strategy.
The next step is a recursive invocation of the optimizer for all entries stored in the priority queue, which corresponds to a depth search (Algorithm 1). If a solution exists, this approach finds always a first result with an upper bound of run time. The maximal number N of nodes to evaluate for a first solution for the number of $n = |\mathbb{B}|$ of bus stops is

$$N = \frac{n \cdot (n+1)}{2} + 1 \tag{8}$$

The complexity for finding a first solution is $\mathcal{O}(n^2)$. Longer run time can improve the result.

Algorithm 1: Optimizer with the specific heuristic

```
Global: function simulate (S)
        function costs (S)
Input:  set E      // existing charging stations
Output: Optimal set of charging stations X
 1: Y ← set of all bus stops
 2: F ← set of bus stops forbidden for charging stations
 3: X ← { }
 4: minCosts ← ∞
 5: function optimize (E)
 6:     priorityQueue pq
 7:     TMP ← E
 8:     // breadth-first search
 9:     for all i ∈ Y\(F ∪ E)
10:         TMP ← TMP ∪ { i }
11:         c ← costs(TMP)
12:         if c < minCosts
13:             f ← simulate(TMP)
14:             if f = 1        // the solution is feasible
15:                 X ← TMP
16:                 minCosts ← c
17:             else
18:                 rank = f/c
19:                 pq.put( rank, TMP)
20:                 pq.truncate(k)
21:             endif
22:         endif
23:         TMP = TMP \ { i }
24:     endfor
25:     // depth-first search
26:     while not pq.isempty
27:         optimize (pq.get( ))
28:     endwhile
29: endfunction
```

CLOSED LOOP PLANNING

The infrastructure-planning algorithm needs a bus schedule and the vehicle schedule planning needs a charging infrastructure. An initial step, which assumes that all bus stops are equipped with chargers, solves this chicken and egg situation. The result is a schedule for an oversized charging environment. With this schedule, the infrastructure can be reduced to a necessary size and the initial costs for all chargers and all buses can be calculated.
The objective function for closed-loop planning takes into account the investments and operating costs for buses and chargers, including all necessary installations. We assume that all buses are of the same type, are equipped with the same battery capacity and have the same price of cb without quantity discounts. Let the price for a charger at bus stop i be $\boldsymbol{cc}_i$ and $\boldsymbol{ss}_i = 1$ if at that stop a charger is required and $\boldsymbol{ss}_i = 0$ otherwise. The global objective is given in (9).

$$cb \cdot |\mathbb{B}| + \sum_{i=1}^{|\mathbb{S}|} \boldsymbol{cc}_i \cdot \boldsymbol{ss}_i \rightarrow min \tag{9}$$

The loop starts after this initialization. Each cycle starts with an infrastructure planning followed by planning a new schedule. If the absolute amount of the costs of the new solution is below a threshold then the loop terminates. Otherwise, the energy demand of the buses is changed in order to get different charging locations and different schedules in the next cycle. This change is applied to the infrastructure planning only. Decreasing the energy demand increases in general the number of chargers and the infrastructure becomes oversized. The new schedule, which is based on such an extended infrastructure, could require fewer buses and the overall costs could decrease.

RESULTS

The introduced algorithms are implemented in pure Java. For first tests, artificial data form a model generator were sufficient. Next results came from investigations with bus operator companies in Germany and in Poland. Current work is on a medium sized model for the city of Braunschweig, Germany. This model consists of 1153 bus stops, 1758 ways and 230 buses.
All optimization were done on a Windows 7 System running on an Intel(R) Xenon(R) Hardware at 2.4 GHz and 8 Mbyte of Memory.

Figure 4 shows the number of charging stations and the costs versus CPU time. In the beginning and after finding a good solution, the run time of the simulator is very often short because of an abort due to the lack of energy. After 90 s a first solution with more than 120 charging devices was found (see Figure 4). For the next 270 s the number of required chargers drops to 25. This value does not decrease during hours.

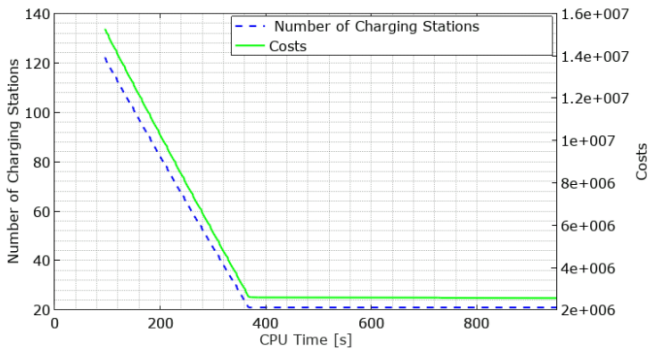


Figure 4: Number of Charging Stations and Costs versus CPU Time

The development of the total costs is in this example very close the number of chargers because they almost all cost the same. The advantage of this optimizer is a calculable upper limit for the time taken to determine the first solution.
Figure 5 shows a bar graph for idle time, time for charging one bus and times for charging buses simultaneously.

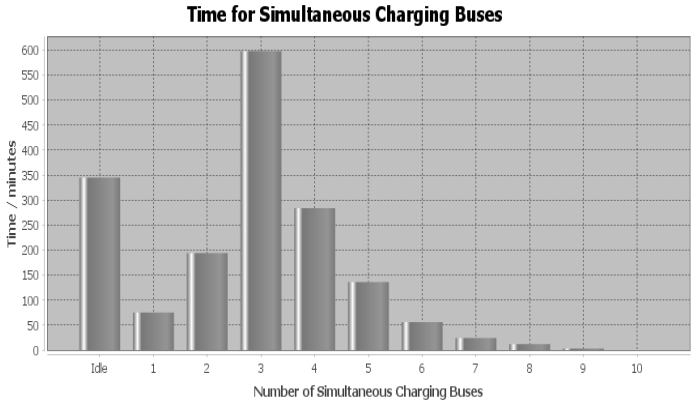


Figure 5: Time for Simultaneous Charging Buses

If we assume that all buses charge with the same power then we can derive the power peaks on the grid directly.

This approach for the optimizer solves the planning problem for charging infrastructures. It runs fast and delivers results for practical use. Nevertheless, some improvements could be made to the optimizer. A dynamic length of the priority queue could keep the solution space small and investigate more candidates of interest. All nodes with feasibility to costs relation (see Algorithm 1, line 18) which are better or close to the already queued best candidates will be put into the queue. An adaptive algorithm should control this dynamic behavior. Another task is parallelizing the optimizer in order to reduce the computing time and to use more than one processor kernel.
The limits of earlier solutions (Buechter and Naumann 2016a; 2016b) are expanded. Now, applications with of more than 1000 buses and more than 10000 bus stops can be solved.
The next step is to test the optimizer and the estimated schedule in a closed loop.

CONCLUSIONS

The problem of planning a charging infrastructure for electrical driven buses is solved by a simulation driven branch and bound algorithm. The simulation controls the selection process and the lower bound is given by an objective function.
For further development, charging on the road with energy supply by pantographs or inductive energy transfer could be integrated in the simulator. As an advantage of the chosen software structure, this does not affect the optimizer. The optimizer only works on sets without any problem specific semantics. The simulator then works with a set of ways instead a set of bus stops.

REFERENCES

Alonso, M.; Amaris, H.; Germain, J.G.; Galan, J.M., 2014, “Optimal Charging Scheduling of Electric Vehicles in Smart Grids by Heuristic Algorithms”, In: Energies (2014)7, Pages 2449-2475, ISSN 1996-1073

Buechter, H. and Naumann, S.. 2016a. “Optimizing Charging Infrastructure for Electrically Driven Buses.” In: European Simulation and Modelling Conference ESM 2016, pages 384–386.

Buechter, H. and Naumann, S. 2016b. “A Hybrid Planning Method for Charging Infrastructure for Electrically Driven Buses in Public Transportation.” In: Grzegorz Sierpiński (Editor) Intelligent transport systems and travel behavior. 13th Scientific and Technical Conference "Transport Systems. Theory and Practice 2016" Katowice, Poland, September 19-21, 2016 : selected papers., Cham: Springer (Advances in intelligent systems and computing, Volume 505), pages 175-185.

Chen, T. D., Kockelman, K. M., Moby Khan, et al. 2013. “The electric vehicle charging station location problem: A parking-based assignment method for Seattle.” In: Transportation Research Board 92nd Annual Meeting, volume 340, 2013, pages 13-1254

Enriquez, M. 2011, "A modular software framework supporting simulation-driven optimization techniques", In: Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th, 2011, pages 7E3-1-7E3-10.

Espinouse, M. L.; Lacomme, P.; Moukrim, A.; Tchernev, N. 2000. "Branch and bound approach based on simulation for the job, input sequencing and device dispatching in FMS" In: IFAK Management and Control of Production and Logistics, Conference Proceedings, Grenoble, pages. 729–734.

Frade, I.; Ribeiro, A.; Goncalves, G.; Antunes, A.P. 2011. “Optimal location of charging stations for electric vehicles in a neighborhood in Lisbon, Portugal.” In: Transportation Research Record, Journal of the Transportation Research Board, 2011, pages 91-98.

Kooten Niekerk van, M. E.; van den Akker, J. M.; Hoogeveen, J. A., 2017. “Scheduling electric vehicles” In: Public Transp (2017) 9, pages 155–176.

Kuchshaus, V.; Bioly, S.; Klumpp, M. 2012. “Deducting E-Mobility Loading Station Locations from City Parking Data.” In: 17th International Working Seminar on Production Economics, Conference Proceedings, Innsbruck 20.-24.02.2012, Bd. 2: Innsbruck, pages. 81–92.

Lam, A. Y. S.; Leung, Y.-W.; Chu, X. 2014. “Electric vehicle charging station placement: Formulation, complexity, and solutions.” In IEEE Transactions on Smart Grid (5), pages 2846-2856.

Olsen, N., Kliewer, N. 2016, “Electric Vehicle Scheduling - A study on charging modeling for electric vehicles.” In: Operations Research Proceedings, 2016, pages 82–102.

Sassi, O; Oulamara, A, 2014a. “Electric Vehicle Scheduling and Optimal Charging Problem: Complexity, Exact and Heuristic Approaches” In: *International Journal of Production Research* (2014) 3, DOI: 10.1080/00207543.2016.1192695

Sassi, O.; Oulamara, A., 2014b. “Joint Scheduling and Optimal Charging of Electric Vehicles Problem” In: *Computational Science and Its Applications – ICCSA 2014*: 14th International Conference, Guimarães, Portugal, Proceedings, pages 76-91

Trumpold, J.; Oertel, R.; Weißer, D.; Schaefer M.; Maier, P; Naumann, S.; Büchter, H., 2017. "MENDEL: Minimum load of electrical networks caused by charging operations of electric buses", In: EVS30 International Battery, Hybrid and Fuel Cell Electric Vihicle Symposium, 9.-11. October 2017, Stuttgart.

Zheng, Y.; Dong, Z. Y.; Meng, K.; Zhao J. H.; Qiu, J., 2014. “Electric Vehicle Battery Charging/Swap Stations in Distribution Systems: Comparison Study and Optimal Planning” In: IEEE Transactions on Power Systems, Vol. 29, No. 1, January 2014, pages 221-229.

SIMULATION IN ENERGY

AN OFFSHORE ENERGY SIMULATION THROUGH FLOW NETWORKS: *CEL* WITHIN THE *MSP CHALLENGE 2050* SIMULATION GAME PLATFORM

Kevin Hutchinson, Harald Warmelink, Wilco Boode, Carlos Santos, Igor Mayer
Academy for Digital Entertainment
Breda University of Applied Sciences
Monseigneur Hopmansstraat 1, 4817 JT Breda,
The Netherlands
E-mail: hutchinson.k@buas.nl

KEYWORDS
Offshore energy simulation, maritime spatial planning, flow network, maximum flow problem, Dinic's algorithm.

ABSTRACT

This paper presents the design of the offshore energy simulation *CEL* as a flow network, and its integration in the MSP Challenge 2050 simulation game platform. This platform is designed to aid learning about the key characteristics and complexity of marine or maritime spatial planning (MSP). The addition of CEL to this platform greatly aids MSP authorities in learning about and planning for offshore energy production, a highly topical and big development in human activities at sea. Rather than a standard flow network, CEL incorporates three additions to accommodate for the specificities of energy grids: an additional node for each team's expected energy, a split of each node representing an object into input and output parts to include the node's capacity, and bidirectional edges for all cables to enable more complex energy grid designs. Implemented with Dinic's algorithm it takes less than 30ms for the simulation to run for the average amount of grids included in an MSP Challenge 2050 game session. In this manner CEL enables MSP authorities and their energy stakeholders to use MSP Challenge 2050 for designing and testing more comprehensive offshore energy grids.

INTRODUCTION

MSP Challenge 2050 (henceforth MSP Challenge) is a novel simulation game platform designed to aid learning about the key characteristics and complexity of marine or maritime spatial planning (MSP). MSP is a process conducted by different governments surrounding a sea basin such as the North Sea, ending up in a spatial plan for each country's areas therein, i.e., their territorial waters and exclusive economic zones. More specifically, it is a process by which a country 'analyse[s] and organise[s] human activities in marine areas to achieve ecological, economic and social objectives' (European Union 2014), ending in a spatial plan. This spatial plan is essentially a highly annotated map of the area with spatial designations for specific human activities and marine protection measures for the medium-term future, often a period of 5-10 years. MSP Challenge was first conceived and developed as a computer simulation game in 2011 and has been applied in sessions with MSP authorities, stakeholders and students many times since (Mayer et al. 2014, 2013; Stolte et al. 2013). Since early 2016 it has been further developed at Breda University of Applied Sciences within the context of the EU projects and consortia *NorthSEE, Baltic LINes* and *SIMCelt*. It has now become a platform allowing for all sorts of simulation game sessions: in different sea basins, with different data sources, and with different simulation models running in the background.

A very topical and big development in human activities at sea is offshore energy production. MSP authorities all over the world are highly concerned with finding sites for the development of offshore renewable energy (mostly wind farms) to help achieve economic and sustainability targets set in various (inter)national policy agreements (Kafas et al. 2018; Borrmann et al. 2018; McGowan 2018). For MSP authorities it has therefore become crucial to dive into their country's overall energy production and the intricacies of offshore renewable energy systems development in the sea basin involved.

At present offshore energy production is a highly complex endeavour involving the development, maintenance and possibly upgrade, and future decommissioning of mostly wind farms. The creation of individual wind farms and their onshore connection to an electricity grid is already complicated, as it involves finding suitable sites (e.g. shallow waters and generally high wind speeds) and the selection of suitable technologies (e.g. turbines, pylons, cables, transformers), while these technologies continue to develop (e.g. bigger turbines, new turbine designs and constraints). The complexity arises when taking into consideration the development of large transnational grids of multiple energy systems such as traditional fossil fuel and wind energy, as well as the diverse consequences to other human activities and marine life.

Simulation gaming greatly helps players understand and deal with a complex endeavour such as offshore energy production (Bekebrede, Lo, and Lukosch 2015; van Bilsen, Bekebrede, and Mayer 2010; Mayer 2016). An offshore energy production simulation game would allow MSP authorities to develop and test offshore energy systems with their energy stakeholders in a safe environment. Moreover, given the spatial relevance, it would be very useful if that simulation game could work within the MSP Challenge

platform. The added value of the MSP Challenge platform is that it already offers the framework of the MSP process, i.e., the process of collectively developing, reviewing, amending and approving one's plans. It also offers the option of planning all sorts of other human activities, such as shipping infrastructure, and marine protection measures, such as no-shipping or no-fishing zones. Within the MSP Challenge platform, players could thus design offshore energy systems in an integrated manner.

In this paper we therefore answer the question *how an offshore energy simulation could be designed and implemented within the MSP Challenge platform, allowing players in a multiplayer setting to spatially plan and implement offshore energy production in an attempt to reach a predefined energy demand or target*.

We answer this question by explaining the process of designing and implementing the offshore energy simulation *CEL* as a flow network. The entire team, that extends beyond the authors of this paper, went through this process over a period of almost a year. The team consulted with key energy experts within the *NorthSEE* and *Baltic LINes* consortia and networks at different stages along the way. We conclude with the main opportunities and limitations that the current design of CEL introduces.

OFFSHORE ENERGY DESIGN IN MSP CHALLENGE

The MSP Challenge simulation game platform has a multi-player client-server architecture, where the server processes inputs from the connected clients, feeds input into any connected simulations (notably an ecosystem simulation), receives output from these simulations, and feeds back data to the connected clients. All simulations running in the background have a discrete-event architecture. Each discrete event represents one simulated month and ideally takes only about a second to run. The time between each discrete event is defined by the facilitator or game master and depends on how long he/she wants the entire session to take. MSP Challenge simulates the planning process in periods of 10 years each, up to a maximum of 40 years, during which players play in country teams to design and implement MSPs, analyse the outcomes and further consequences, and make new plans accordingly.

The addition of an energy simulation is meant to allow MSP Challenge players to design more comprehensive energy production plans, within the wider MSP context. MSP Challenge players have always been able to designate areas for energy production such as wind farms, and then connect them to shore via an electricity cable. With this next development step, we wanted to have the platform calculate how much energy the players are actually generating, transporting and consuming with their energy production areas.

To simplify this both for the players and the development team, we do not consider the network as a whole but divide it into separate grids. Grids consist of sources that generate power, cables and transformer stations that transport power, and sockets that consume power. Each grid is an independent part of the network, meaning they are not connected and can in no way influence each other. Each grid's energy output is added up and fed back to the players as the total amount of energy created in their network. Like all spatial designations in MSP Challenge, each grid element (e.g. a wind farm or electricity cable) can belong to a different team. This allows teams to not only co-develop a grid, it also forces them to specify how much power each team adds to or receives from the grid.

Determining the energy distribution is part of the grid design process. These energy distributions only specify what different teams expect to get from a grid, not what they actually get. Players may set the expected amount of energy to a large amount, such as the maximum amount that a wind farm can generate. However, if the cables or transformers in the grid cannot handle this, the energy they actually receive will be lower. Determining and placing cables and transformers that allow the right amount of energy to pass through without wasting too much capacity is a big part of the challenge for the player.

Players create these grids in the game client. The grid designs are then sent to and stored on the game server. When the game's time progresses another month, the game needs to know how much energy teams get from their grids. This is where the energy simulation CEL comes in, which is the focus of the remainder of the paper.

CEL'S ARCHITECTURE

In this section we explain the CEL architecture by specifying what we feed into it as input, what we would like to get out of it as output, and why we chose to approach the simulation as a flow network.

Specifying CEL's input: grids

As explained, CEL's input is a set of individual grids drawn by MSP Challenge teams defining the entire energy network. We need to define of what a grid consists more formally to ensure that the flow of energy through them can be simulated, both technically and realistically. Grids consist of four parts:

1. Input: Sources, such as wind farms
2. Output: Sockets
3. Medium: Cables and transformers
4. Distribution: A list of expected outputs per team that is part of the grid.

Each of these elements can only be in a single grid at a time. When a cable is added between two grids, their contents are combined and they become a single grid. We resolve the energy simulation for each grid separately and then combine the results to present the total amount of energy produced and consumed.

Sources, sockets, cables and transformers are all objects in the world. They are limited by a capacity that is specified in the configuration of the game (thus not in CEL, but in the MSP Challenge platform). When creating a grid, the relevant teams need to specify how much energy they expect to get, and thus how the energy is distributed. The energy that teams expect to get from a grid is not simply a fixed value that they purely set themselves. The value is subject to two rules:

1. Each team's expected energy has to be lower than or equal to the sum of the capacity of their sockets in the grid.
2. The sum of all teams' expected energy combined needs to be lower than or equal to the total energy available in the grid.

Note that the medium is not present in these rules. The expected energy specifies what teams would get if the medium was perfect, i.e., has the right capacity to carry all produced energy all the way to the socket. In practice the medium might not accommodate the necessary amount of energy to reach all teams' expected values.

Specifying CEL's output: energy KPIs

Having defined more formally what type of data CEL receives as input, we can now specify CEL's output. To define the output we first consider what data we want to display to the players in the game client. We feed back the following variables to the teams as key performance indicators of the energy grids and overall energy network:

- All objects (cables, transformers, sockets and sources) displaying the amount of energy that passed through them.
- An overview per grid, showing how much energy all teams participating in the grid received from it.
- An overview per team and per game session, showing the amount of energy produced, consumed and shared.

To show this information the minimum output we need from CEL is the amount of energy that passes through all objects in the network. All other data can be derived from this. However, it would be easier in programming terms if the output also includes how much energy each team got per grid.

A flow network

The CEL design is based on the simulation objective: *given a grid, find the maximum amount of energy that can flow from the sources to the sockets, with each team's sockets collectively limited by the output expected by the team as defined in the grid design.*

The first part of this phrasing reveals our design foundation – considering energy grids as flow networks – rendering our problem similar to the maximum flow problem (Ahuja 2017; Schrijver 2002). Indeed, others have also approached energy grids as flow networks (Fang et al. 2018). If we can represent these particular energy grids as flow networks, there are a host of existing algorithms to calculate the maximum flow.

However, there are several important differences between a traditional flow network and an energy grid:

- Traditional flow networks have one source and one terminal (or sink), while grids can have multiple sources and sockets (when conceived as terminals).
- Traditional flow networks are directed graphs, and therefore have unidirectional edges, while energy cables (when conceived as edges) are bidirectional.
- Nodes in traditional flow networks do not have limited capacities, while transformer stations, sockets and sources in energy grids do (when conceived as nodes).

DESIGNING CEL AS A FLOW NETWORK

In this section we discuss how we chose to conceive our energy grids as flow networks, thereby resolving the aforementioned issues.

Initial representation

Let us see how we can represent an energy grid as a flow network step by step. Grid A (Figure 1) is used as an example throughout this section. It consists of two sources generating 11 energy units in total that are connected to a transformer, which is in turn connected to two sockets. The capacities of all elements are indicated. In this example we assume that the two teams specified the distribution of the 11 energy units as follows: Team 1 gets five energy units and Team 2 gets six units. Upon further examining the capacities in this example, in practice Team 1 could indeed get its requested five units, but then Team 2 would only get three units. This is how we want the system to work.

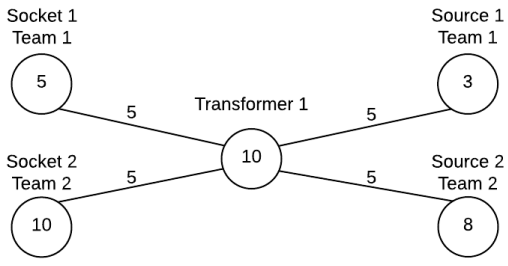


Figure 1: Example Grid A

If we convert this directly to a flow network, we get what is represented in Figure 2. The source *s* and terminal *t* were added, all edges became directional, and all nodes (energy sources, transformer and sockets) lost their inherent capacity. The capacity of the energy sources and sockets are used as the capacity of the edge connecting them to the source and terminal. The capacity of the transformer is lost,

which is a key problem. If we were to run this network in a maximum flow algorithm, the results would probably not match our expected outcomes.

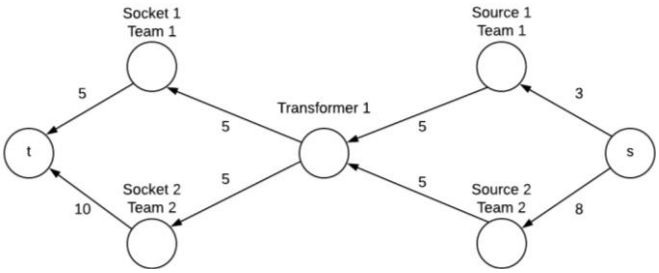


Figure 2: Initial Flow Diagram of Grid A

Allowing for energy distribution

The flow network of Figure 2 is not properly taking the energy distribution set by the two teams into account. The distribution essentially specifies a combined capacity for all sockets of a team in the grid. In the example both teams only have a single socket. Therefore the problem could be solved by setting the capacity of the edges to the terminal to the expected value of that socket's team. However, if a country had multiple sockets this approach would not work, as the flow network would not be able to keep track of the total amount of energy that each team is expecting. To accommodate this option, an intermediary node of each team's expected energy is added (Figure 3).

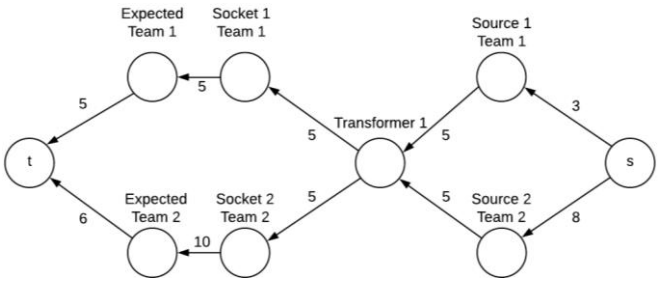


Figure 3: Grid A Limited by Energy Distribution

Introducing node capacity

The energy sources can receive a limited amount of energy from the flow network's source, and the energy sockets can send a limited amount of energy to the terminal. However, how much energy flows through the nodes themselves is not limited in our example flow network. As already mentioned this is particularly a problem for the transformers.

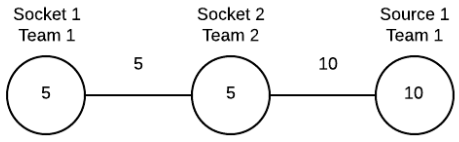


Figure 4: Example Grid B

Consider Grid B (Figure 4), a new example where the distribution of the 10 energy units is as follows: Team 1 gets five energy units and Team 2 gets five units. In our currently proposed design we would have to represent this with the flow network of Figure 5.

In Figure 5 both teams would receive the total amount of five energy units they each expect (at the nodes Expected Team 1 and 2 respectively). However, Socket 2 would be noted as passing on 10 energy units (rather than only receiving that amount), even though its capacity is only five units. This same problem applies to energy sources; they could pass on more energy than their capacity should allow.

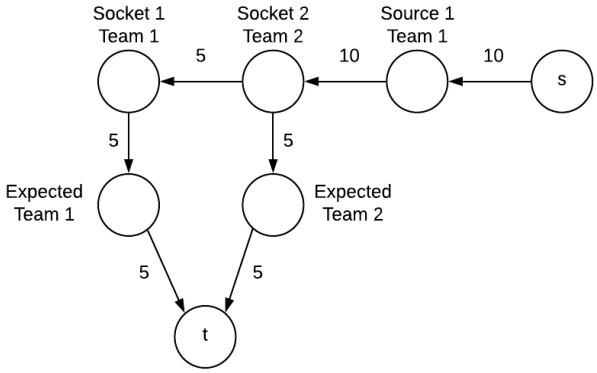


Figure 5: Grid B as a Flow Network

Adding node capacity solves this particular problem. To add node capacity to the flow network we split all nodes into two parts: input and an output. The input part of the node has an edge to the output part with the capacity of the original node. Figure 6 shows how Grid B would be affected by the introduction of node capacity to the flow network.

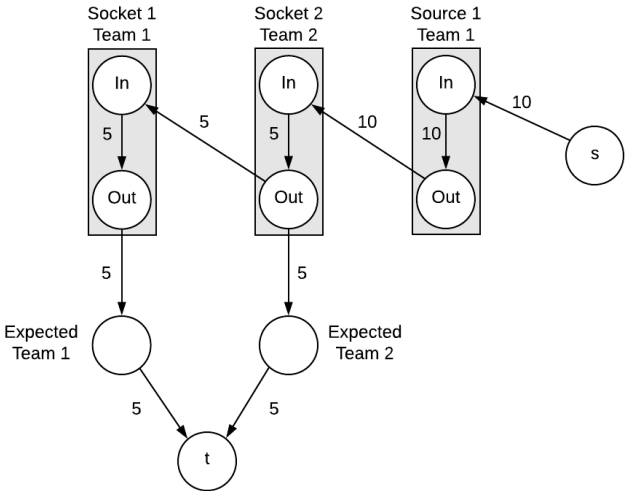


Figure 6: Grid B with Split Nodes

Applying node capacity on our original Grid A leads to Figure 7. This figure combines the added node of each team's expected energy to accommodate for energy distribution (Figure 3) with the socket's input-output distinction for node capacity.

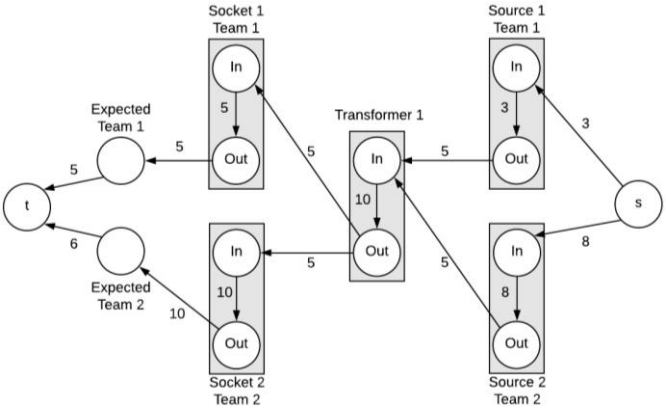


Figure 7: Grid A with Split Nodes

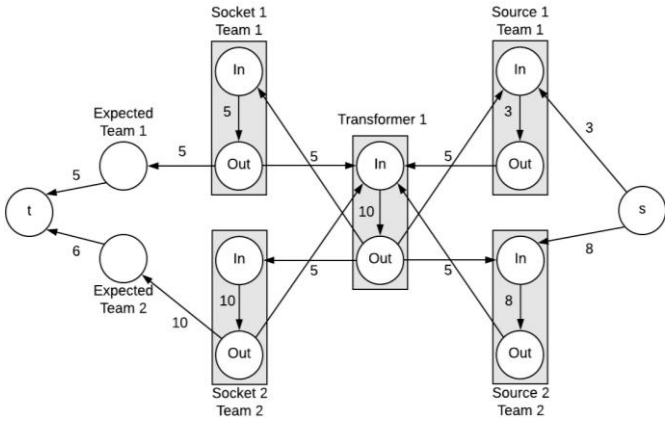


Figure 8: Grid A with Bidirectional Edges

Introducing bidirectional edges

In an energy grid the maximum flow can be achieved by only using directional edges. This is because energy will never flow through a cable in both directions at the same time. Of course, in different time frames energy can pass through a cable in different directions. One could thus argue that all cable edges should be bidirectional. Yet from a flow network perspective, even when an edge is bidirectional, energy will only ever pass through it in one direction at the same time.

This is why, until now, we represented our cables with unidirectional edges and determined this direction beforehand. This was not a problem in the example of Grid A, since it was obvious in what direction energy would pass through the cable. However, when grids become more complex and contain loops, predetermining directionality becomes more complex. Instead of trying to determine directionality beforehand, we simulate bidirectional edges and let the maximum flow algorithm determine which way to send the energy through it.

To accomplish this we duplicate the edges representing cables, with the duplicate having the same capacity but a reversed direction. Because the nodes are split into two parts, this reversed edge would go from an input node to an output node, which is incorrect. The new edge should therefore be moved to go from the output node of its origin to the input node of its destination. Applying this to Grid A results in Figure 8.

In summary

With these changes we can now represent energy grids as flow networks while still retaining the important properties of an energy grid:

1. To limit the energy per team to the value they have defined in their grid design, we added an additional node of expected energy per team, connecting their sockets to the terminal with an edge with a capacity of their expected value.
2. To represent node capacities we split all the nodes representing objects into input and output parts, connected with an edge with the node's capacity.
3. To represent bidirectional edges we duplicated all the edges representing cables and reversed their direction.

The created flow network can now be used as input for a maximum flow algorithm to calculate the flow of energy through a grid per simulated month in MSP Challenge.

CEL IN PRACTICE

Path prioritization

When putting the presented design into practice, a final consideration that has to be made is how energy is distributed among the appropriate teams if their expected values cannot be reached. In other words: how much does each team actually get when none of their expectations can be met? There are two main options to handle this situation:

1. The energy is spread equally.
2. It is undefined.

The first option has a lot of additional considerations. How is ‘equally’ defined? What happens if energy cannot be distributed equally? Do we prioritize an equal distribution over the maximum flow? This option can also not be implemented by changing the the flow network design. It requires changing the maximum flow algorithm applied.

For MSP Challenge the second option was deemed appropriate. If players plan well, their expected values will be reached. If not, the maximum flow algorithm will simply do its work, determine that the maximum flow does not equal the expected flow and maximum capacity of the energy sources, and present results based on whatever paths the algorithm ended up with. CEL feeds back the number of energy units that every object passes on. Thus the residual energy can be calculated and fed back to the players as 'wasted' energy. This information is an incentive for players to analyse and revise their grid design as part of their MSP, serving the very purpose of MSP Challenge.

Performance

The presented additions to the flow network design naturally decreases performance when compared to a direct conversion to a flow network (as posed in Figure 2). We should note that this is not a fair comparison, because a direct conversion would not adhere to the properties of an energy grid. Nevertheless, the comparison can give us an indication of the additional performance cost incurred.

The performance of maximum flow algorithms is usually expressed in E and V, where E is the number of edges and V the number of vertices (nodes). Our additions to the standard flow network change these values into E' and V' in the following ways, where T is the number of teams in the grid:

$$E' = 2E + T + V \quad (1)$$
$$V' = 2V + T \quad (2)$$

In MSP Challenge Dinic's algorithm (Dinic 1970) was used for its robustness and good performance. Dinic's algorithm has a complexity F (equation 3). With our additions the complexity F' needs to be calculated as shown in equation 4.

$$F = O(V^2 \ E) \quad (3)$$
$$F' = O((V + T)^2 \ (E + T + V)) \quad (4)$$

In practice, with the average amount of grids in MSP Challenge, the energy simulation takes less than 30ms to run a simulated month. This includes the data requests done to the server. We note that we tested the solution on high-end hardware.

CONCLUSION

In this paper we presented how a dive into the complexities of offshore energy production with the help of experts from the NorthSEE, Baltic LINes and SIMCelt consortia and networks led to an amended flow network simulation called CEL. We chose to represent offshore energy grids as flow networks, with three important additions: an additional node for each team's expected energy, a split of each node into input and output parts to include node capacity, and bidirectional edges for all cables to enable more complex energy grid designs. Implemented with Dinic's algorithm, we have an energy simulation CEL that is suitable for use by MSP authorities and their energy stakeholders within the simulation game platform MSP Challenge. CEL's performance is at such a level, that it can be safely and usefully incorporated into the MSP Challenge platform.

With this design CEL enables MSP authorities and their energy stakeholders to use MSP Challenge 2050 for designing and actually testing more comprehensive offshore energy grids. An additional value of our approach is that we can also use it for other energy infrastructures, notably fossil fuel energy production (offshore oil and gas notably). An advanced use of MSP Challenge could thus concern the decommissioning of fossil fuel energy production balanced with the development of offshore renewable energy production. This way MSP Challenge would be used for offshore energy transition management.
We note several limitations in our approach. Our energy simulation is still a simplification of an offshore energy system. The simulation also does not take external influences on energy flow into account, such as fluctuating wind speeds. We note, however, that we can deal with the latter outside CEL by letting MSP Challenge dynamically configure the maximum capacity of an energy source with each discrete event (thus each simulated month). Still, we have to keep in mind that our goal is not to offer a complete offshore energy electrical engineering design system. MSP authorities need to know enough about the electrical engineering involved to come up with comprehensive and theoretically feasible offshore energy production designs befitting relevant policy objectives and within the wider MSP context. We believe the presented CEL design helps achieve that.

FUTURE RESEARCH

Our next step is to apply CEL in offshore energy MSP Challenge sessions within the North Sea and Baltic Sea regions. We aim to help these regions' MSP authorities and energy stakeholders plan for offshore energy in an integrated manner, thus also considering other human activities as well as the impacts and protection measures for marine life. These sessions are part of the NorthSEE and Baltic LINes projects and will take place at the end of 2018 and beginning of 2019.

The question remains how and to what extent MSP Challenge and CEL will aid MSP processes in the different sea regions we apply them. Moreover, as the MSP Challenge platform continues to develop, the question remains whether and how we should adjust or optimize CEL to improve performance or enable new features. A different maximum flow algorithm could help improve performance, if needed. Related to this question, is the matter of overall MSP Challenge development and support methodology. How should the platform be maintained, further developed and used, and by whom? We aim to answer these questions in the coming years.

ACKNOWLEDGEMENTS

The research leading to these results acknowledges the contribution of the NorthSEE, Baltic LINes and SIMCelt projects, which respectively received funding from the European Union's Interreg North Sea Region (NSR) and Baltic Sea Region (BSR) programs (both part of the European Regional Development Fund), and the European Maritime and Fisheries Fund (EMFF).

REFERENCES

Ahuja, R. K. 2017. *Network Flows: Theory, Algorithms, and Applications*. 1st ed. Pearson Education.

Bekebrede, G.; J. Lo; and H. Lukosch. 2015. "Understanding Complex Systems Through Mental Models and Shared Experiences." *Simulation & Gaming* 46, No.5, 536–62. https://doi.org/10.1177/1046878115621463.

Bilsen, A. van; G. Bekebrede; and I.S. Mayer. 2010. "Understanding Complex Adaptive Systems by Playing Games." *Informatics in Education* 9, No.1, 1–18.

Borrmann, R.; K. Rehfeldt; A-K. Wallasch; and Silke Lüers. 2018. "Capacity Densities of European Offshore Wind Farms." http://www.vasab.org/index.php/documents/doc_download/1581-capacity-densities-of-european-offshore-wind-farms.

Dinic, E A. 1970. "An Algorithm for the Solution of the Max-Flow Problem with the Polynomial Estimation." *Doklady Akademii Nauk SSSR* 194, No.4, 1277–80.

European Union. 2014. "Directive 2014/89/EU of the European Parliment and of the Council of 23 July 2014 Establishing a Framework for Maritime Spatial Planning." *Offical Journal of the European Union* 2014 (Apr), 135–45.

Fang, J.; C. Su; Z. Chen; H. Sun; and Per Lund. 2018. "Power System Structural Vulnerability Assessment Based on an Improved Maximum Flow Approach." *IEEE Transactions on Smart Grid* 9, No.2, 777–85. https://doi.org/10.1109/TSG.2016.2565619.

Kafas, A; M. Ripken; K. Wright; M. Billet; S. Sangiuliano; E. Ooms; and U. Scheffler. 2018. "Status Quo Report on Offshore Energy Planning Provisions in the North Sea Region." http://www.northsearegion.eu/media/4930/northsee-offshore-energy-status-quo-main-report-final-version-120418.pdf.

Mayer, I.S. 2016. *Playful Organisations & Learning Systems*. Breda: NHTV Breda University of Applied Sciences. https://www.researchgate.net/profile/Igor_Mayer/publication/304816825_Playful_Organisations_Learning_Systems/links/577bc2eb08ae213761cab129/Playful-Organisations-Learning-Systems.pdf.

Mayer, I.S.; Q. Zhou; X. Keijser; and L. Abspoel. 2014. "Gaming the Future of the Ocean: The Marine Spatial Planning Challenge 2050." In *Serious Games Development and Applications*, 150–62. Lecture Notes in Computer Science. Springer, Cham. https://doi.org/10.1007/978-3-319-11623-5_13.

Mayer, I.S.; Q. Zhou; J. Lo; L. Abspoel; X. Keijser; E. Olsen; E. Nixon; and A. Kannen. 2013. "Integrated, Ecosystem-Based Marine Spatial Planning: Design and Results of a Game-Based, Quasi-Experiment." *Ocean & Coastal Management* 82 (Supplement C): 7–26. https://doi.org/10.1016/j.ocecoaman.2013.04.006.

McGowan, L. 2018. "Offshore Wind: Maritime Sector Briefing Note." http://www.simcelt.eu/wp-content/uploads/D2c_Offshore-Wind-Briefing-Note.pdf.

Schrijver, A. 2002. "On the History of the Transportation and Maximum Flow Problems." *Mathematical Programming* 91, No.3, 437–45. https://doi.org/10.1007/s101070100259.

Stolte, W.; X. Keijser; A. de Kluijver; and A. J. Nolte. 2013. "A Conceptual Pressure-Response and a Simple Food Web Model for the Maritime Spatial Planning Challenge 2050 Serious Game." 1208167-000-ZKS-0007. Delft: Deltares.

WEB REFERENCES

www.balticlines.eu
www.mspchallenge.info
www.northsee.eu
www.simcelt.eu

BIOGRAPHIES

Kevin Hutchinson, Wilco Boode and Carlos Santos are all part of the Breda University of Applied Sciences (BUAS) games and media research and development team. Kevin Hutchinson functions as a core developer of MSP Challenge, while Wilco Boode functions as the lead designer, and Carlos Santos as the technical manager. Harald Warmelink is a senior researcher at BUAS, and functions as project manager of MSP Challenge. Igor Mayer is professor of serious and applied gaming at BUAS, and functions as the project leader of MSP Challenge.

NON COOPERATIVE GAME THEORETIC APPROACH FOR RESIDENTIAL ENERGY MANAGEMENT IN SMART GRID

Ilyes Naidji
Faculty of Mathematical, Physical and Natural Sciences,
University of Tunis-El Manar, Tunisia.
Email: ilyes.naidji@fst.utm.tn

Moncef Ben Smida
LSA Lab, Tunisia Polytechnic School,
University of Carthage, Tunisia.
Email: moncef.bensmida@gmail.com

Mohamed Khalgui
National Institute of Applied Sciences and Technology,
University of Carthage, Tunisia.
Email: khalgui.mohamed@gmail.com

Abdelmalik Bachir
LESIA Lab, Computer Science Department,
Biskra University, Algeria.
Email: a.bachir@univ-biskra.dz

KEYWORDS
Energy management, Demand response, Discomfort level, Game theory, NSGA-II

ABSTRACT

Demand side management (DSM) is one of the main functionalities of the smart grid as it allows the consumer to adjust its energy consumption for an efficient energy management. Most of the existing DSM techniques aim at minimizing the energy cost while not considering the comfort of consumers. Therefore, maintaining a trade-off between these two conflicting objectives is still a challenging task. This paper proposes a novel DSM approach for residential consumers based on a non-cooperative game theoretic approach, where each player is encouraged to reshape its electricity consumption pattern through the dynamic pricing policy applied by the smart grid operator. The players are guided to select the best strategy that consists of scheduling their electric appliances in order to minimize the daily energy cost and their discomfort level. The Nash Equilibrium of the energy management game is achieved using Non-Sorting Genetic Algorithm NSGA-II. Simulation results show the effectiveness of the distributed non cooperative game approach for the residential energy management problem where an appreciable energy cost reduction is reached while maintaining the discomfort in an acceptable level.

INTRODUCTION

Along with the current transition of the power system from a centralized to a distributed architecture (Mosbahi and Khalgui 2016), a great attention is being paid to the power grid's capacity to maintain the balance between demand and supply (Meskina et al. 2017, Abidi et al. 2017). Indeed, the increasing penetration of distributed renewable energy sources (RES) which have an intermittent nature induces new challenges for the smart grid in terms of energy management, congestion, voltage and frequency variations, etc. In order to overcome these challenges, the demand side management (DSM) is increasingly exploited by smart grid operators to maintain the demand-supply equilibrium taking advantage of the demand flexibility. DSM brings many solutions for consumers such as energy saving through the reduction of the electricity consumption and best use of electric appliances. Recent advances in information and communication technologies offer the opportunity for advanced DSM solutions, e.g., demand response, time of use, spinning reserve, etc.

In this study, we investigate the demand response (DR) solutions to reduce the energy cost for consumers while ensuring their comfort. The DR solutions consist of the short-term changes in the power consumption that could be made in response to the energy price variation. The dynamic pricing is designed to incite consumers to participate in the DSM by decreasing or increasing their power consumption. In addition, DR solutions do not only consist in reducing the power consumption but can modify the consumption pattern. DR is enabled through communication infrastructures (Fadel et al. 2015), allowing to decrease energy consumption during peak periods. It has been shown that DR can solve some existing problems in traditional power systems and enhance the reliability (Safamehr and Rahimi-Kian 2015).Smart grid technologies enable another pricing methods in restructured form. The prices are variable with respect to the demand and operating conditions, which involve the consumers participation in the power system operation. Various methodologies have been proposed for the energy management of the smart grid using demand response. In (Al Zahr et al. 2017), advanced demand response is proposed considering modular and deferrable loads with the objective of reducing the cost of consumed energy and peak consumed power. In (Kaddah et al. 2014), direct load control (DLC) programs have been proposed. DLC allows the energy provider to directly control (switch on/off) the electric appliances of the consumers with respect to their agreements (e.g., maximum number of interruptions, appropriate re-

wards, etc). Simply shifting the power consumption of consumers at a peak time to off-peak times may cause consumer's discomfort. To model the discomfort of consumers, the difference between the desired load and the scheduled load is considered in (Deng et al. 2014).When shifting the power consumption pattern, the difference between the desired and scheduled time of loads can be considered to evaluate consumer's discomfort. In (Eksin et al. 2015), a game theoretic approach was used to solve the demand response using the Bayesian Nash equilibrium with the objective of minimizing the peak-to-average ratio. However the comfort level of consumers was not addressed. In (Ning et al. 2017), a coordinated optimization is proposed where the concept of demand response potential (DRP) was introduced. However, the approach did not verify the consumer's comfort with concrete result.

Most of the aforementioned studies focus on the reduction of the energy cost and peak load to solve the energy management problem and do not sufficiently consider consumer behavior. In particular, the comfort level of consumers was not considered simultaneously with the energy cost in the demand side studies. Even when jointly considered with the energy cost like in (Kim et al. 2013), they are subsequently referred as a total cost. Such consideration may affects the result of the energy management system and may give a biased solution sometimes for the energy cost and sometimes for consumers comfort.

In this respect, we propose a multiobjective demand response game for the optimal scheduling of the electric appliances in smart homes. Non cooperative energy management game is designed to model the behavior of consumers. The proposed energy management game incorporates the demand side in the supply management using the dynamic pricing policy to minimize daily energy cost while reducing the discomfort level of consumers with a multiobjective approach.

The contributions of this paper can be summarized as follows:

- We propose a non cooperative energy management game which guarantees the fairness among non-cooperative consumers and apply the NSGA-II algorithm to find the Nash equilibrium of the game in accordance to the two conflicting objectives that are energy cost and consumer's comfort.
- We model the rational behavior of the consumers that reduce their energy cost while seeking their comfort.
- We address the concept of discomfort level of consumers which is based on the difference between the desired and scheduled time of the electric appliances.

This paper is organized as follows: Section gives the smart grid model used in this paper. Section explains the concept of game theory for the energy management problem. Section gives the formulation of the proposed energy management game. Section gives the numerical results. Finally, Section concludes this paper.

SYSTEM MODEL

This section describes the smart grid system used in this study and gives the energy management system (EMS) architecture. Consider a smart distribution grid with one electricity provider that supplies energy to a set of n consumers, i.e., smart homes which are controlled by a Multi-Agent System (MAS). Each smart home is equipped with a smart meter that is connected to the electric appliances via wire connection e.g., PLC (Power Line Communication) or wireless connection, e.g., ZigBee, etc. Furthermore, the smart meter integrates an agent that has computational intelligence capabilities. Each agent collects the planned tasks for the current day and power consumption profiles of the electric appliances from the smart meter. Consumers communicate with the smart meter through wireless communication, e.g., smart phones or tablets, to indicate the preferred conditions of their electric appliances related to their comfort, e.g., the preferred temperature in the rooms, desired time to charge the electric vehicle, etc. Fig. 1 shows the architecture of the proposed EMS.

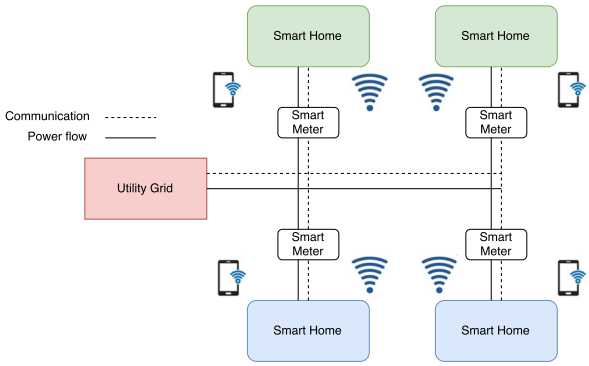


Figure 1: Distributed EMS Architecture.

In the proposed EMS, each consumer is characterized by its planned daily tasks TS_j. Let $TS = \{TS_1, TS_2, ..., TS_k\}$ be the set of daily tasks to be executed, each task can be characterized by two vectors (Salinas et al. 2013):

$$X_j = \left[x_j^1, x_j^2, ..., x_j^t\right] \quad (1)$$

$$Y_j = \left[P_j \quad D_j \quad ST_j \quad FT_j \quad STP_j \quad FTP_j\right] \quad (2)$$

where

- X_j is the power consumption profile of the electric appliance that executes the task j where x_j^t is the power consumption of task j at time t.

- Y_j is the vector which characterizes the task j.
- $P_j = \sum_{t=1}^{T} x_j^t$ is the energy demand of the task j where the time horizon $T = 24$.
- D_j is the duration of the task j.
- $[ST_j,\ FT_j]$ are the earliest start time and finishing time to run the task j that define its admitted interval of execution.
- $[STP_j, FTP_j]$ is the time preferred window of the consumer to run the task j.

Pricing method

Let P_{ij}^t is it the energy consumption of consumer i for task j at time t. The total energy consumption of all consumers $(i = 1, ..., n)$ at time slot t is defined as follows:

$$l^t = \sum_{i=1}^{n}\sum_{j=1}^{k} P_{ij}^t \tag{3}$$

Consider $C_u^t(l^t)$ the cost function of utility grid at time slot t. The quadratic cost function is usually used in literature (Deng et al. 2014), i.e,

$$C_u^t(l^t) = a^t.(l^t)^2 + b^t.l^t + c^t \tag{4}$$

where the quadratic costs coefficients (a^t, b^t, c^t) are time varying. Real-time pricing is used in this study where the price value is time varying. The price value depends on time of use (TOU) and total energy consumption. Based on this pricing model, the energy cost of consumers at time slot t is defined as in (Deng et al. 2014) by:

$$C_c^t = c_r^t(l^t).\sum_{j=1}^{k} P_j^t \tag{5}$$

where $c_r^t(l^t)$ is the real-time price of energy. We assume that the smart grid operator adopts adequate pricing method that takes into consideration the energy consumption in time and level. In this study, we consider also consumers within distributed energy resources (DERs) facility that can produce energy. The revenue of the consumer R_c^t is calculated as follows:

$$R_c^t = \sum_{t=1}^{T} r.P_g^t \tag{6}$$

where P_g^t is the power generated from the consumer at time t and r is the revenue coefficient.

Consumer discomfort level

To measure the discomfort of consumers caused by shifting their consumption pattern, we introduce a discomfort cost as a quadratic function of the gap between the desired and the scheduled time of the electric appliances. We define a time shift parameter Δ_j that models the gap between the scheduled and preferred time of the task j. Let t_j denotes the start time of the task j. The time shift parameter Δ_j is calculated as follows:

$$\Delta_j = \begin{cases} 0, & \text{if } t_j \geq STP_j \wedge t_j + D_j \leq FTP_j, \\ STP_j - t_j, & \text{else if } t_j \leq STP_j \wedge t_j + D_j \leq FTP_j, \\ (t_j + D_j) - FTP_j, & \text{else if } t_j \geq STP_j \wedge t_j + D_j \geq FTP_j, \end{cases} \tag{7}$$

Here, for each task j, the concept of time shift can be valid only in the admitted interval of execution $[ST_j, FT_j]$ of the task j. The discomfort cost can be modeled with a quadratic cost function (Samadi et al. 2012) as follows:

$$C_{\Delta_j} = \sum_{j=1}^{k} \alpha(\Delta_j)^2 + \beta\Delta_j + \delta \tag{8}$$

where α, β and δ are the quadratic cost coefficients. Here, as more as the time shift parameter increases, i.e., the electric appliance is scheduled out of its preferred time, the discomfort cost increases.

MATHEMATICAL GAME THEORY FORMULATION

This section gives the mathematical game theory formulation of the energy management problem involving the MAS.

In the energy management problem, the consumers do not collaborate, for example, when a smart meter shows the real-time electricity price in the smart grid, the consumer reduces or increases its electricity consumption without asking neighbors whether they reduce their consumption or not at a certain time. Game theory is able to model the competitive behavior of the consumers. In the proposed energy management architecture, each consumer, i.e., agent is a player of the non-cooperative game. The energy management game can be defined by the following 3-tuple:

$$G = \{N, S, J\} \tag{9}$$

where N is the set of players with $|N| = n$. S is the strategy space of players where

$$S = S_1 \times S_2 \times ... \times S_n \tag{10}$$

$J = S \rightarrow \mathbb{R}$ is the vector of cost functions of players $i = 1, 2, ..., n$ which is defined as

$$J(s) = [J_1(s), J_2(s), ..., J_n(s)] \quad s \in S \tag{11}$$

where the vector of strategies $s = (s_1, s_2, ..., s_n) \in S$ is called a strategy profile. Let $\{a_i\}$ be the set of actions of the player i. Each action a_i represents the total energy consumption of player i over the time slot t. For

player i, the set of the selected actions consists of the energy consumption pattern for the time horizon $T = 24$, i.e., one day. The strategy s_i of the player i can be regarded as a rule for choosing its actions. For example, the player, i.e., consumer can use the remaining energy in the car battery to cover the expensive evening consumption and re-charge his car battery after midnight. Moreover, consumers within DERs that are covering their energy needs and having surplus, can re-sell the energy to the grid, even using their own home storage facilities and recharging them when the energy is cheaper. The cost function of the player i is given by

$$J_i(s) = J_i\{s_i^*, \overline{s_i^*}\} \tag{12}$$

where the cost function J_i depends on the strategy $s_i^* \in S_i$ selected by the player i and on the strategy profile $\overline{s_i^*}$ of the other players. Solving the energy management game consists of finding the Nash Equilibrium for each player in the non cooperative game. The vector $s^* = (s_1^*, s_2^*, ..., s_n^*)$ is a Nash equilibrium for the energy management game $G = \{N, S, J\}$ if the following constraint is valid:

$$\forall i \in N, \forall s_i \in S_i, \quad J_i(s_i^*, \overline{s_i^*}) \leq J_i(s_i, \overline{s_i^*}) \tag{13}$$

PROBLEM FORMULATION

This section describes the energy management game and the considered game players.

Energy management game players

The proposed energy management game consists of two types of players that are: consumer denoted by (c-player) and consumer within DERs that can produce energy denoted by (p-player). C-player represents the flexible consumers that can adjust their consumption pattern. P-player represents the consumers within DERs facility. Each player has its own objective functions, as illustrated in the following.

C-player
The c-player can adjust its consumption pattern through managing its smart electric appliances. The first objective of the c-player is to reduce the energy cost as follows:

$$\min \quad J_1 = \sum_{t=1}^{T} C_c^t \tag{14}$$

The second objective of the c-player is to minimize its discomfort level by

$$\min \quad J_2 = \sum_{j=1}^{k} \alpha(\Delta_j)^2 + \beta\Delta_j + \delta \tag{15}$$

Hence, the cost function of the c-player is given by

$$J_c(s) = (J_1, J_2) \tag{16}$$

P-player
The p-player can adjust its consumption pattern and manage its DERs. The first objective of the p-player is to minimize its energy cost and maximize its revenue as follows:

$$\min \quad J_3 = \sum_{t=1}^{T} C_c^t - R_c^t \tag{17}$$

where C_c^t, R_c^t are the energy cost and revenue of p-player at time slot t, respectively.
The second objective of the p-player is to minimize its discomfort level by

$$\min \quad J_4 = \sum_{j=1}^{k} \alpha(\Delta_j)^2 + \beta\Delta_j + \delta \tag{18}$$

Thus, the cost function of the p-player is given by

$$J_p(s) = (J_3, J_4) \tag{19}$$

Constraints

The proposed energy management game is subject to the following constraints.

Time constraints
Each task j of duration D_j must be executed exactly once between its earliest start time ST_j and finishing time FT_j. The start time t_j of task j satisfies the following constraint:

$$ST_j \leq t_j \leq FT_j - D_j \tag{20}$$

Energy balance
The energy generated by the utility grid and p-player must be equal to the total energy consumed by c-player and p-player satisfying:

$$E_u^t + E_p^t - E_d^t = 0 \tag{21}$$

where E_u^t is the energy produced by the utility grid, E_p^t is the energy produced by the p-player and E_d^t is the total energy demand of all consumers, i.e., c-players and p-players.

Nash equilibrium
The existence of the Nash equilibrium is proved by the following theorem (Nash 1951):

Theorem 1. *Every game with a finite number of players that can choose from finitely number of strategies has at least one Nash equilibrium.*

In the proposed non cooperative game, each player is assumed to be rational, i.e., the player aims to minimize its cost function by considering the best strategy. Therefore, each player chooses the load profile that represents its best strategy.

Theorem 2. *The combination of best strategies and their corresponding cost functions constitutes the dominant strategy which is the Nash equilibrium for the energy management game (Fudenberg and Tirole 1991).*

NSGA-II scheduling

NSGA-II algorithm is used to find the Nash equilibrium of the energy management game. A detailed description of the algorithm can be found in Deb et al. (2002). Each player, i.e., agent gets the information about time-differentiated electricity price from the smart grid operator to adjust the scheduling of its daily tasks. After that, the agent applies the NSGA-II algorithm to solve the multiobjective optimization problem with the objective to minimize the daily energy cost and consumer's discomfort. Hence, the NSGA-II algorithm specifies the best strategy of each player in accordance to the minimization of two fitness functions. For the c-player, the strategy is chosen by

$$s_i^{*c-best} = \arg\min \left[J_c(s)\right] \tag{22}$$

For the p-player, the strategy chosen by

$$s_i^{*p-best} = \arg\min \left[J_p(s)\right] \tag{23}$$

The NSGA-II scheduling solution gives the dominant strategy for each player which is the best strategy. The combination of these best strategies constitutes the dominant strategy which is the Nash Equilibrium of the energy management game (Fudenberg and Tirole 1991).

SIMULATION RESULTS

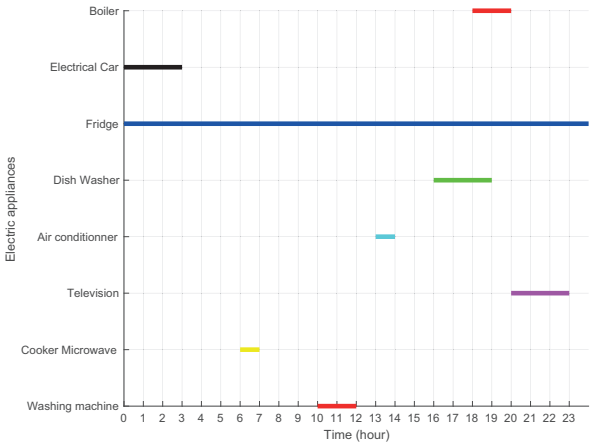
This section illustrates the system under study and gives the simulation results of the proposed multiobjective energy management game.

For testing the proposed method, the EMS is developed in MATLAB software. The system under study consists of a smart grid with $|N| = 30$ consumers. The system includes several c-players and p-players. Each player performs its daily tasks, at least 8 tasks selected randomly from Table 1. To obtain the Nash equilibrium of the energy management game, our proposed multiobjective formulation was submitted to NSGA-II algorithm.

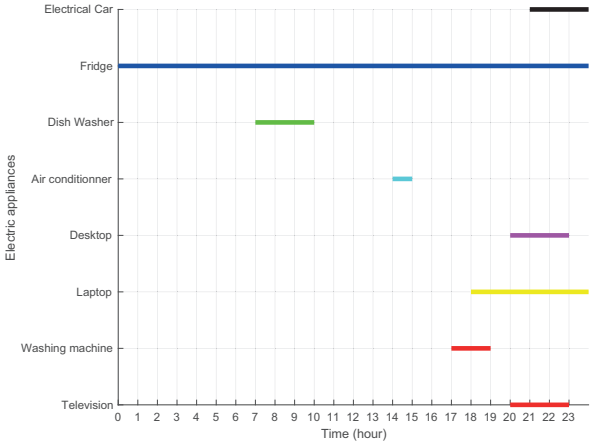
In order to incentivize the consumer to run some of its electrical appliance at particular time-slots of the day, the smart grid operator uses time-variable tariff rates. Fig. 2 summarizes the set of considered electricity appliances and highlights the starting and finishing time of these appliances that can be scheduled in the admitted time window. The first subfigure shows the appliances of a typical c-player and the second subfigure shows the appliances of a typical p-player.

Table 1: Task's characteristics.

Task	Power (kW)/h	ST_j (hour)	FT_j (hour)	Duration (hour)
Washing machine	1	6	24	2
Laptop	0.1	18	24	6
Desktop	0.3	18	24	3
Air conditionner	1.5	10	19	1
Dish Washer	1	7	19	3
Fridge	0.3	0	24	24
Electrical Car	3.5	18	8	3
Boiler	0.8	15	22	2
Iron	1.2	10	22	1
Cooker Microwave	1.7	6	9	1
Spin Dryer	2.9	13	18	1
Television	0.6	19	24	3
Cooker Oven	5	18	19	0.5
Cooker Hob	3	8	9	0.5



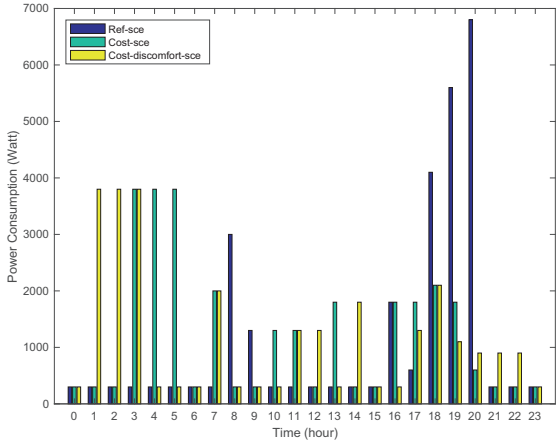
(a) C-player.



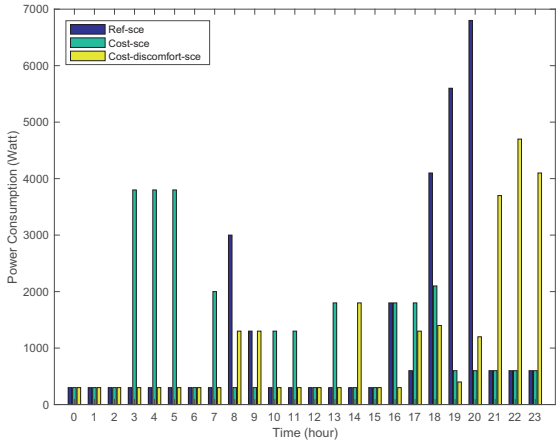
(b) P-player.

Figure 2: Tasks scheduling.

To evaluate the effectiveness of the proposed scheduling solution, three scenarios are investigated. The first is the reference scenario ('Ref-sce'), where each task will be executed at its preferred time interval and will not be executed in early or later time. In the literature, such scenario is called the welfare maximization as in (Li et al. 2011). The second is the cost effective scenario ('Cost-sce') as in (Al Zahr et al. 2017), where the tasks are executed with the objective to minimize the daily energy cost without taking into account the discomfort of consumers. The third scenario ('Cost-discomfort-sce') refers to the proposed multiobjective scheduling solution which consists of the minimization of the daily energy cost and discomfort level of consumers.
Fig. 3 shows the total power consumption for a typical c-player and a typical p-player for the considered scenarios, while Figs. 4 and 5 show the results of the considered scenarios for all the consumers in terms of energy cost and discomfort level.



(a) C-player.



(b) P-player.

Figure 3: Power consumption.

It can be seen that p-player has more comfort than c-player taking the advantage of its power generation, which reduces its energy cost, such that, the p-player schedules its electric appliances during its preferred time. On the other hand, c-player has less comfort due to the absence of the power generation, such that, the consumer schedules its electric appliances out of its preferred time.

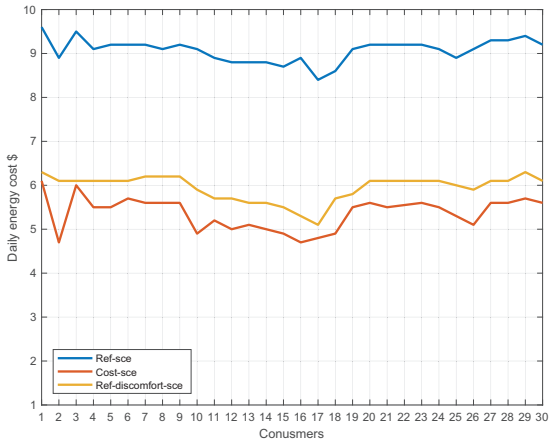


Figure 4: Daily energy cost.

As expected, 'Ref-sce' has the highest power consumption. It is observed during peak hours from 18:00 to 20:00. 'Ref-sce' has also the highest daily energy cost of electricity and with discomfort level equal to zero. A significant decrease in the power consumption is observed in the 'Cost-sce' from 18:00 to 20:00.

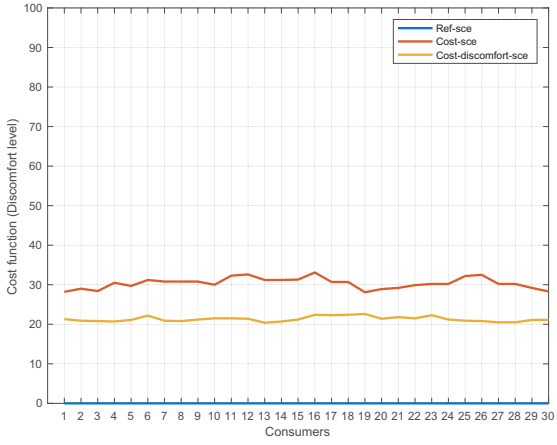


Figure 5: Discomfort level.

It also achieves the lowest daily energy cost with a reduction of about 40% compared with the 'Ref-sce'. However, discomfort level increased by at most 30%. The 'Cost-discomfort-sce' has a low power consumption where the reduction in the daily energy cost is about 37% compared with the 'Ref-sce'. However, the daily

energy cost increased by at most 3% compared with the 'Cost-sce'. The discomfort level is maintained in an acceptable level, which is increased by 20% compared with the 'Ref-sce' and decreased about 10% compared with the 'Cost-sce'.
In summary, the proposed task scheduling strategy achieved its main objective which is a tradeoff between shifting the power consumption to time-slots where the daily energy cost is cheaper and maintaining the discomfort of consumers in an acceptable level, thanks to the proposed multiobjective game formulation.

CONCLUSION

In this paper, a non cooperative game theoretic approach has been proposed to implement a DR energy management strategy for competitive residential consumers to obtain a minimum daily energy cost and minimum discomfort level. The method presented has been solved using NSGA-II algorithm. In the proposed game theoretic approach, the consumers play a key role in the energy management game through exploiting DERs and scheduling their electric appliances locally, in contrast to centralized energy management systems that use DLC approaches that directly control the loads and impose load shedding to the consumers. The developed energy management game can be used as a useful tool for evaluating the electricity market and also for analyzing the strategic behavior of consumers in competitive electricity markets. As a perspective, we will consider additional choices of game theoretic approaches for DSM and more comprehensive decision-making models for consumers based on behavioral sciences.

REFERENCES

Abidi M.G.; Smida M.B.; Khalgui M.; Li Z.; and Wu N., 2017. *Multi-agent oriented solution for forecasting-based control strategy with load priority of microgrids in an island mode–Case study: Tunisian petroleum platform. Electric Power Systems Research*, 152, 411–423.

Al Zahr S.; Doumith E.A.; and Forestier P., 2017. *Advanced Demand Response Considering Modular and Deferrable Loads under Time-Variable Rates.* In *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 1–6.

Deb K.; Pratap A.; Agarwal S.; and Meyarivan T., 2002. *A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE transactions on evolutionary computation*, 6, no. 2, 182–197.

Deng R.; Yang Z.; Chen J.; Asr N.R.; and Chow M.Y., 2014. *Residential Energy Consumption Scheduling: A Coupled-onstraint Game Approach. IEEE Transactions on Smart Grid*, 5, no. 3, 1340–1350.

Eksin C.; Deliç H.; and Ribeiro A., 2015. *Demand Response Management in Smart Grids with Heterogeneous Consumer Preferences. IEEE Transactions on Smart Grid*, 6, no. 6, 3082–3094.

Fadel E.; Gungor V.C.; Nassef L.; Akkari N.; Malik M.A.; Almasri S.; and Akyildiz I.F., 2015. *A survey on wireless sensor networks for smart grid. Computer Communications*, 71, 22–33.

Fudenberg D. and Tirole J., 1991. *Game theory, 1991. Cambridge, Massachusetts*, 393, no. 12, 80.

Kaddah R.; Kofman D.; and Pioro M., 2014. *Advanced Demand Response Solutions Based on Fine-grained Load Control.* In *Intelligent Energy Systems (IWIES), 2014 IEEE International Workshop on*. IEEE, 38–45.

Kim B.G.; Ren S.; Van Der Schaar M.; and Lee J.W., 2013. *Bidirectional energy trading and residential load scheduling with electric vehicles in the smart grid. IEEE Journal on Selected Areas in Communications*, 31, no. 7, 1219–1234.

Li N.; Chen L.; and Low S.H., 2011. *Optimal demand response based on utility maximization in power networks.* In *Power and Energy Society General Meeting, 2011 IEEE*. IEEE, 1–8.

Meskina S.B.; Doggaz N.; Khalgui M.; and Li Z., 2017. *Multiagent framework for smart grids recovery. IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47, no. 7, 1284–1300.

Mosbahi O. and Khalgui M., 2016. *New solutions for optimal power production, distribution and consumption in smart grids. International Journal of Modelling, Identification and Control*, 26, no. 2, 110–129.

Nash J., 1951. *Non-cooperative games. Annals of mathematics*, 286–295.

Ning J.; Tang Y.; Chen Q.; Wang J.; Zhou J.; and Gao B., 2017. *A bi-level coordinated optimization strategy for smart appliances considering online demand response potential. Energies*, 10, no. 4, 525.

Safamehr H. and Rahimi-Kian A., 2015. *A cost-efficient and reliable energy management of a micro-grid using intelligent demand-response program. Energy*, 91, 283–293.

Salinas S.; Li M.; and Li P., 2013. *Multi-objective optimal energy consumption scheduling in smart grids. IEEE Transactions on Smart Grid*, 4, no. 1, 341–348.

Samadi P.; Mohsenian-Rad H.; Schober R.; and Wong V.W., 2012. *Advanced demand side management for the future smart grid using mechanism design. IEEE Transactions on Smart Grid*, 3, no. 3, 1170–1180.

POWER SYSTEM PLANNING SUPPORTED BY BIG DATA

Samuel Marrero Vera, José Évora Gómez, José Juan Hernández Cabrera
University of Las Palmas de Gran Canaria (ULPGC), Spain
Institute of Intelligent Systems and Numerical Application in Engineering (SIANI)
samuelmarrero@siani.es jose.evora@siani.es josejuanhernandez@siani.es

KEYWORDS
Big Data, SCADA, Power System Operation, Flexibility, Event Driven Architectures

ABSTRACT

Power systems, especially those which are small and isolated, are working more and more to the limits of their capacity. Thus there is a need for new structures that give them more flexibility. To this end, the Spanish Transmission System Operator (TSO), Red Electrica de España (REE), in collaboration with various research centres, is working on solutions to increase the flexibility of isolated power systems. The evaluation of these solutions, based in new devices and operational strategies, requires new analysis tools. This article presents one such tool for integrating and comparing simulation and operation results for the system using a solution based in Big Data.

CONTEXT

Power systems are increasingly operating in stress situations. The deregulation of related activities has lead to them being operated in situations approaching their maximum capacity [1]. Moreover, the rise in Variable Generation (VG), which is more difficult to predict than demand, has increased uncertainty in the management of these systems. This situation demands new operational strategies and devices to increase flexibility [2]. At the same time, these will also facilitate the penetration of renewable energy and improve supply security.

The flexibility of a power system can be defined as its capacity to deploy available resources in order to deal with changes in Net Demand (ND) [3]. In the case of power systems that are isolated and weakly meshed, such as those on islands, small changes in ND can cause major disruptions in the system, making its operation much more complicated [4].

The most recent decade has seen the development of new strategies to increase flexibility. This was intended to facilitate the penetration of renewable energy sources while keeping the security of the supply. On the one hand, the main source of power system flexibility is the generators' power reserve [5]. In this respect, Independent System Operators (ISO's) have created new ways to measure and increase flexibility from the generation point of view [5].

On the other hand, power systems also play an important role in providing flexibility. For this reason, whenever it is necessary to study the flexibility of a power system, it is advisable to take into account the limitations imposed by the grid [6].

Traditionally, these limitations have been overcome by incorporating power lines or generators that contribute more transmission and/or generation capacity. Nevertheless, they can also be overcome without adding new systems by integrating FACTS devices. FACTS (Flexible Alternating Current Transmission Systems) are power electronics devices that are able to generate/absorb reactive power in order to control voltage within the electrical transmission network [7].

It was under these circumstances that the project OSMOSE (Optimal System-Mix of flexibility Solutions for European electricity) arose [8], promoted by Transmission System Operators (TSO's), electrical companies and European universities. This project has been integrated into the European Union's Horizon 2020 programme and is studying how to combine sources of flexibility with the objective of taking on more renewable energy.

As part of this project, the Spanish TSO, Red Electrica de España (REE), proposes to demonstrate the effectiveness of a Multi-Component Flexibility System (MCFS) in the power system on the islands of Lanzarote and Fuerteventura (Canary Islands, Spain). The MCFS consists of various flexibility devices operated jointly. These devices are a flywheel and a Hybrid Flexibility Device (HFD) which is composed of a FACTS device and storage devices: batteries and super-capacitors.

In order to evaluate the effectivenes of the flexibility solution implemented, the performance of the MCFS is being analysed. To accomplish this, simulations will be carried out and the grid will be monitored in real time during its operation before and after its implementation. The data obtained is then used to make a comparative analysis.

POWER SYSTEM MANAGEMENT

Power systems are formed by generation units and distribution networks that exchange power via transmis-

sion networks. The management of these systems could be considered from two perspectives: service management and infrastructure management.

Service Management

This is generically referred to as *Operation* and is carried out by the Independent System Operator (ISO). The objective is to achieve a reliable functioning of the system at a minimal cost. In this context, it becomes necessary to develop different activities in distinct time scales [9]:

- *Control*. This involves maintaining the dynamic balance of the system in an instantaneous manner. The devices in charge of this task react automatically to correct any possible discrepancies [9].
- *Operation*. Basically, this involves adjusting the references of the control devices to assure the correct functioning of the power system [9], maintaining the system's dynamic trajectory..
- *Programming*. Given that it is not sufficient to respond instantaneously to counteract changes produced within it, it is also necessary to envision the resources and actions needed to take into account unforeseen behaviours before the operation takes place [9].

To support the service management of power systems, SCADA's (Supervisory Control And Data Acquisition [10]) are used. SCADA's are composed of Remote Terminal Units (RTU's), a communications network and a control centre. In each node of the network there is an RTU, which monitors and controls several devices and electrical lines. RTU's are in charge of sending data and receiving orders according the operation defined by the control centre [10].
In order for the power system to function properly, production must match consumption precisely and instantaneously. Therefore, coordination between the production system and the transmission network assures that the energy produced by the former will be transmitted to the distribution networks. For this to take place, in conformance with the quality standards specified by existing regulations, it is necessary to confront the technical restrictions that the system imposes, limiting their capacity to cope with changes in ND [6].

Infrastructure Management

Carried out by the Transmission System Operators (TSO's), the objective of this activity is to assure the present and future availability of the network to attend to the needs of the service. Management of the infrastructure includes its maintenance, as well as as its development and expansion. The objective is to assure that the system will operate correctly in the future.

In order to satisfy the requirements of energy demand, systems rely on a certain number of power plants and transmission lines [9]. Traditionally, planning the expansion of a power system meant calculating the number of power plants and transmission lines that must be constructed in order to meet future power and energy demands, while minimising investment and operation costs.
Given the importance of adequately planning the expansion of these systems, there are numerous mathematical optimisation models that may resolve this problem. In general, these models help to determine the type, size and location of new power plants and transmission infrastructure that should be installed. The purpose of these tools is to insure the satisfaction of future demand in a way that total investment in new equipment and the expected system operational costs will be minimal.
Nevertheless the changes that the electrical sector is undergoing has created a need for a new infrastructure and operational methods that increase flexibility [11]. For example, the rise in renewable energy generators in recent years has made necessary to take into account flexibility devices in planning studies.
Because the best way to operate this new infrastructure including MCFS is still uncertain, traditional optimisation models are no longer adequate to carry out planning studies.

HYPOTHESIS

Considering that traditional tools are no longer adequate for planning the expansion of the power systems, this paper proposes the following hypothesis:

> The evaluation of a power system which implies service management, requires a tool that can analyse and compare different configurations using data obtained through simulations and monitoring.

Configuration is understood to mean the definition of an infrastructure and its operational system. For this purpose, a tool is proposed to monitor and control multiple simulated configurations in the same way SCADA does with real-life systems. As a result, it will be possible to make comparative evaluations of configurations associated with the installation of new devices by applying a specific operational strategy. Moreover, this will permit comparison with the operation of the real-operation system.
With the aim of knowing how the MCFS device can contribute to maximum flexibility, a set of simulations will be carried out. These basically consist of calculating the voltage in the nodes and the power flow in the power lines.
In order to carry out these simulations, software simulators that are specific to this domain such as PSS/E®

[12], will be used to model the power systems being studied. Models consist of parameterised objects representing each one of the devices and lines belonging to the power system. The idea is that these objects will be reusable for different simulations, which is why it is necessary to parameterise them. Finally, the model will be calibrated to insure its performance corresponds to reality. To this end, data from the real-operation of the system will be used, which should coincide with the simulation results.
Once this model is completed, variants of it will be created to represent different configurations: the settings of the MCFS itself, changes in the distribution network, in generation, in the topology of the grid and the operation itself. In this case, the main variations in the initial models will consist of the inclusion of the MCFS in all its possible variants.
As for the proposed planning tool, there is the possibility of automating the creation of these variants: the parameters of the objects in the model could be changed automatically, thus generating new configurations. As a result, an enormous number of simulations could be conducted.
In the end, each configuration will be simulated in order to generate information that must be analysed and compared. This will allow us evaluate the improvement of flexibility obtained from the MCFS settings, as well as from its operation in different configurations. It will enable determining the best possible MCFS settings for a specific operation strategy and the best planning for flexibility. Finally, once that MCFS is up and running, simulations will be compared with the actual operation of the system.
These simulations will certainly create a huge amount of data as they must be done for all possible operation conditions: fluctuations in demand or multiple contingencies that might affect the operation of the system.
Evaluating these configurations, even in a small power system, requires processing a large amount of data. Considering an analysis of 200 possible configurations with only 15 substations and 5 power plants, performing simulations of a period of 30 days, the result would exceed 500 gigabytes of data. Moreover, new operation methods could produce an even greater flow of data given that there is available data which has not yet been dealt with.
Dealing with this amount of data requires new software architecture based in Big Data [13].

BIG DATA ARCHITECTURE

In order for a SCADA to function as a Big Data system, a set of technologies enable conducting calculations over huge sets of data. For example, Hadoop [14] or MapReduce [15] are used to carry out batch processing on large volumes of data taking advantage of their capacity to perform parallel computing. In such cases, this procedure can take minutes, or even hours.
In normal situations, this type of calculations may be adequate for evaluating the performance of power systems. However, there are situations in which it is necessary to make decisions immediately. For instance, in the operation, it is needed other types of technology that allow stream processing in real time in order to control the devices directly and guarantee the grid's stability. Technologies that can be used to perform this processing include Spark[16], Storm[17] and Data Torrent RTS[18]. For this reason, Big Data solutions involve combining two types of processing; batch and stream, seeking the advantages that each one offers.
It was precisely with the idea of implementing systems that combine both methods of processing data that the Lambda Architecture was created [19]. In concrete, this paper proposes the application of the Lambda architecture to evaluate the flexibility of power systems.
The Lambda architecture is a type of Event-Driven Architecture. In these type of architectures [20], an *event* is a representation of any significant change detected in an entity or any action performed by an actor. Event-driven architectures are based in three software design patterns: Publisher/Subscriber [21], Event Sourcing [22] and Command Query Responsibility Separation (CQRS)[23].
In these types of architectures, all events have a timestamp, from which it is possible to generate time series. This type of representation makes possible to know all the transitions that provoke an entity to arrive at a concrete state. Moreover, it is possible to re-generate them to make sure the results are correct. These architectures include the following components:

- *Event Bus.* It allows publishers to receive generated events and route them to subscribers, as well as register them in the Event Store.

- *Event Store.* This warehouse contains a representation of what has happened to the relevant entities in the domain. In order to do this, it is necessary to define the type of events which it can receive and the format which each one has. In this way, a sort of contract is established with the sensors, which makes it possible to establish not only the structure of the data, but also its semantics.

- *Event Feeders.* Sensors and actors generate events and publish them in a *topic*, which other applications that might be interested in this type of event may subscribe to.

- *Event Handlers.* Applications which want to receive the events arriving to the bus may subscribe to topics. The bus allows the subscribers to define the flow of events they are interested in, in order to generate Domain Models.

- *Domain Models.* Events are projected and accumulated over models in order to perform queries. Models can be reconstructed, re-generating the flow of events again. This allows constructing present, as well as past states.

Specifically, the Lambda architecture takes into consideration the problems of latency, performance and failover, while it is balancing the processing. In order to do this, responsibilities are organised into three layers:

1. *Batch layer* in which a process is conducted in order to provide pre-computed views.
2. *Speed layer* in which a processing of event flow in real time is carried out in order to provide dynamic views.
3. *Serving layer* in which views are indexed and queries are attended to, considering the requisites of the domain latency.

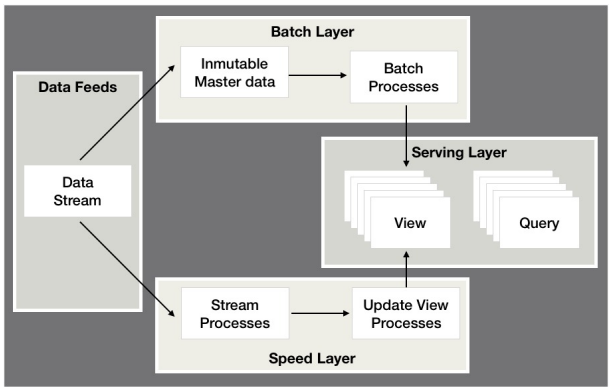


Figure 1: Lambda Architecture

SOLUTION DESIGN

As we have already stated, the design of the solution is based in the Lambda architecture and adapted in order to respond to the challenges faced: evaluating different configurations in the power system in operation. The proposed design can be seen in figure 2.
In the same way as SCADA's on real power systems, simulations have virtual RTU's which allow sending data to the devices in operation, as well as controlling them. These RTU's also send the measurements to the control centre for their processing.
Simulation tools are considered sources of events that feed the Datalake. That is to say, simulations generate prospective contexts of the power system. This means that the data coming out of these simulations must be treated in the same way than data from the RTU's in a real power system.
These measurements will be transformed into events that will feed both the Datalake (event store) and the

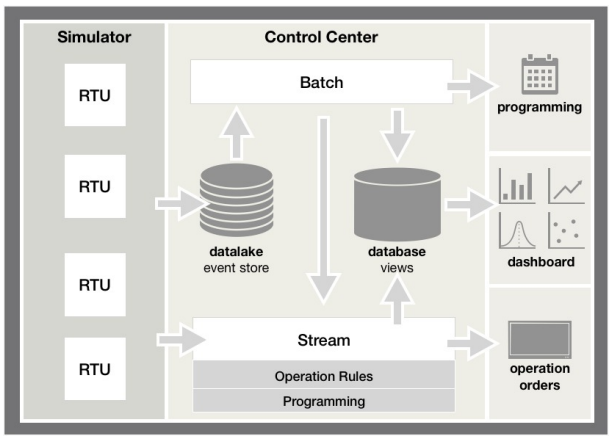


Figure 2: OSMOSE Lambda Architecture

speed layer. The events that arrive at the Datalake in the batch mode will support the construction of different models to be used for evaluation and calculation of indicators. These models, according to their nature, will offer analysis capacity that will allow evaluation of the grid's performance.
The intention is that the real-life system, as well as the simulations, will create analysis contexts that will enable comparing different simulation configurations with each other or with data from real-life systems.
In any of these contexts, events will not only be stored in the Datalake to be processed in the batch layer, but will also be processed in the speed layer by a rule engine. This rule engine will evaluate if the grid is functioning correctly and carrying out corrective operations. These corrective operations could be performed automatically, in the case of simulations, or function as an alert system so that the grid operator can make appropriate decisions. The operation of power systems in real time should be done in a matter of minutes [9], which requires including a set of rules in the speed layer in order to analyse the system quickly.
These rules could verify that the corrective measures applied are functioning correctly, following a cycle of measurement-decision-action. That is to say, events that arrive to rule system via the speed layer help in making decisions; these decisions can lead to the performance of a corrective action; the effects of this corrective action will appear in the following measurements which arrive to the rule system.

DISCUSSION

The objective of the OSMOSE project consists of designing and evaluating a flexibility solution for the islands of Lanzarote and Fuerteventura based on a MCFS device. In order to do this, a SCADA system with an orientation to Big Data has been proposed. This would allow integrating data also from simulations to carry out a

comparative analysis between them.
It should be noted that the proposed architecture respects the existing infrastructure for acquiring data, and this will reduce the impact of the solution on data systems already in place. On the other hand, it opens new gateways with simulation tools for the compilation of data and virtual operations performed in simulations. This is achieved by integrating the simulations in a SCADA that also allows comparing different configurations. In order to evaluate these configurations, indicators used by the system operators for service management are calculated and compared.
The main advantage of this proposal is that it allows creating a repository of events for planning activities, both in real-life and simulated systems.
This will be achieved by integrating systems in a SCADA, which will also allow comparing different configurations. To evaluate these configurations, indicators used by the system operators managing the services will be calculated and compared. These indicators will be used to establish comparisons between the different configurations: reality vs simulation, before the MCFS and after the MCFS.
It also would led support to evaluate decisions made by operators. That is to say, the programming and operation of the system could be evaluated through simulations that operate like a traditional SCADA.
To this end, the tool will allow adjusting operation procedures, including the possibility of "reversing" events. Checkpoints could be established, through which different decisions could be analysed. This creates the possibility of analysing different operation strategies for the system. This requirement is important given that it is necessary to find out how a power system should be operated using an MCFS device.
However, this is not the only advantage of this proposal. It could also function for the operation of the system, given that it already respects the infrastructure for acquiring existing data. Moreover, predictive models implemented in the simulation tools, which function as data sources, could be included in the operation.
Finally, the availability of time series and operation results opens the possibility of introducing intelligent operation models based on Machine Learning.

CONCLUSION

In this article we have presented a tool for planning power systems based on Big Data architecture design principles. This tool has been designed and developed in the context of the OSMOSE project, giving support to the evaluation of different flexibility devices that are going to be installed. The tool is proposed as a SCADA that can coexist with other SCADA's already installed. The design is based on an event-driven software architecture: The Lambda architecture, applying different design principles such as Event Sourcing, CQRS and Publisher-Subscriber. In the Lambda architecture, data flow is handled in two streams: one is stored to be processed later, the other is processed in real time.
This proposed solution allows processing time-series of events, in both batch and stream modes. Likewise, the tool allows analysing different configurations, which correspond with data sent by the RTU's, as well as data generated during simulations. Moreover, two configurations associated with the real-life system are created: before and after the installation of the MCFS.
In this initial phase, this tool is foreseen for use in the planning the infrastructure, but for later phases we propose functions that would allow also be permitted in the real-life operation. That is to say, to be used in the management of both the infrastructure and the system.

Acknowledgement. This paper is part of the project OSMOSE which has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement N° 773406

References

[1] Belkacem Mahdad, Tarek Bouktir, and Kamel Srairi. "Strategy of location and control of FACTS devices for enhancing power quality". In: *IEEE Mediterranean Electrotechnical Conference* (2006).

[2] N. Menemenlis, M. Huneault, and A. Robitaille. "Thoughts on power system flexibility quantification for the short-term horizon". In: *IEEE Power and Energy Society General Meeting* (2011). DOI: 10.1109/PES.2011.6039617.

[3] Eamonn Lannoye, Damian Flynn, and Mark O'Malley. "Evaluation of power system flexibility". In: *IEEE transactions on power systems* 27.2 (May 2012). DOI: 10.1109/TPWRS.2011.2177280.

[4] Helena Vasconcelos et al. "Advanced control solutions for operating isolated power systems". In: *IEEE Electrification Magazine* 3 (Feb. 2015). DOI: 10.1109/MELE.2014.2380131.

[5] Eamonn Lannoye, Damian Flynn, and Mark O'Malley. "Power system flexibility assessment - State of the art". In: *IEEE Power and Energy Society General Meeting* (2012). DOI: 10.1109/PESGM.2012.6345375.

[6] Eamonn Lannoye, Damian Flynn, and Mark O'Malley. "Transmission, Variable Generation and power system flexibility". In: *IEEE Transactions on power systems* 30.1 (Jan. 2015). DOI: 10.1109/TPWRS.2014.2321793.

[7] Laszlo Gyugyi Narain G. Hingorani. *Understanding FACTS: concepts and technology o Flexible AC Transmission Systems*. IEEE, 2006. ISBN: 978-0780334557.

[8] OSMOSE. URL: https://cordis.europa.eu/project/rcn/213922_en.html.

[9] Alexandra von Meier. *Electric power systems: a conceptual introduction*. Ed. by Wiley-Interscience. IEEE, 2006. ISBN: 978-0-471-17859-0.

[10] A.P. Sakis Meliopoulos. "Power system modeling, analysis and control". In: *Georgia Institute of Technology* (2006).

[11] Juan Ma et al. "Evaluating and planning flexibility in sustainable power systems". In: *Power and Energy Society General Meeting (PES), 2013 IEEE*. IEEE. 2013, pp. 1–11.

[12] PSS®E. URL: https://www.siemens.com/global/en/home/products/energy/services/transmission-distribution-smart-grid/consulting-and-planning/pss-software/pss-e.html.

[13] Vangelis Marinakis et al. "From big data to smart energy services: an application for intelligent energy management". In: *Future generation computer systems* (2018). URL: https://doi.org/10.1016/j.future.2018.04.062.

[14] Apache Hadoop. URL: http://hadoop.apache.org/.

[15] Jeffrey Dean and Sanjay Ghemawat. "MapReduce: simplified data processing on large clusters". In: *Communications of the ACM* 51.1 (2008), pp. 107–113.

[16] Apache Spark. URL: https://spark.apache.org/.

[17] Apache Storm. URL: http://storm.apache.org/.

[18] Datatorrent RTS. URL: https://mapr.com/apps/datatorrent/.

[19] Nathan Marz and James Warren. *Big Data: Principles and best practices of scalable real-time data systems*. New York; Manning Publications Co., 2015.

[20] Brenda M Michelson. "Event-driven architecture overview". In: *Patricia Seybold Group* 2.12 (2006), pp. 10–1571.

[21] Martin Fowler. URL: https://martinfowler.com/eaaDev/EventCollaboration.html.

[22] Martin Fowler. URL: https://martinfowler.com/eaaDev/EventSourcing.html.

[23] Martin Fowler. URL: https://martinfowler.com/bliki/CQRS.html.

SIMULATION OF THE LB-LOCA IN CANDU 6 REACTORS WITH RELAP/SCDAPSIM(IUA). MODELLING EFFECTS OF THE BREAK LOCATION

Roxana-Mihaela Nistor-Vlad
Daniel Dupleac
Ilie Prisecaru
Politehnica University of Bucharest
Splaiul Independenţei 313, Sector 6
060042, Bucharest, Romania
E-mail:
roxanamihaelanistorvlad@gmail.com,
danieldu@cne.pub.ro, prisec@gmail.com

Marina Perez-Ferragut
Chris Allison

Innovative Systems Software
3585 Briar Creek Ln
Ammon, ID 83406, USA
E-mail: marinotxka@gmail.com,
iss@srv.net

KEYWORDS

CANDU, LOCA, uncertainty, RELAP/SCDAPSIM, modelling effects.

ABSTRACT

RELAP/SCDAPSIM is a flexible nuclear tool designed to describe the thermal hydraulic response and core behaviour under normal operating conditions, design basis accidents or severe accident conditions. An Integrated Uncertainty Package was developed by Innovative Systems Software (ISS) jointly with Polytechnic University of Catalunya (UPC), and the uncertainty evaluation capability is implemented as an alternative run mode, the "uncertainty" mode, which allows the automatic execution of an uncertainty analysis based on the probabilistic approach. A complete uncertainty analysis using RELAP/SCDAPSIM/MOD3.4 code requires the execution of three related phases, namely the "setup" phase, the "simulation" phase consisting of several executions, and the "post-processing" phase.

A Large Break Loss Of Coolant Accident (LB-LOCA) is defined conventionally for CANada Deuterium Uranium (CANDU) reactors as the rupture of a Primary Heat Transport System (PHTS) component where the break area is larger than twice the cross sectional area of the largest feeder pipe.

This paper will describe the CANDU 6 behaviour during a postulated LB-LOCA event with the uncertainties considered.

INTRODUCTION

A LB-LOCA in a CANDU is characterized by a break in the PHTS which initiates reactor shut down. There is an initial period of blowdown as the pressurised coolant vents from the system. As the coolant pressure drops, cooling is degraded and the fuel undergoes a temperature transient. During this period, there is the possibility for the fuel assemblies to be damaged and for the fission products to be released in the containment. The Emergency Core Cooling System (ECCS) plays an important role to the accident progression. The initiation of the cold coolant injection by the ECCS refloods the reactor core (fuel channels), maintaining the fuel assemblies well cooled and termiatig the accident. The unavailability of the ECCS could turn into a severe accident, with severe core damage.

The main purpose of this paper is to observe break location modelling effects on the relevant parameters of the plant during a LB-LOCA accident in a CANDU 6 reactor when uncertainties are accounted.

INTEGRATED UNCERTAINTY PACKAGE

Recently, an uncertainty package has been implemented in the RELAP/SCDAPSIM thermalhydraulic and safety analysis tool. This package performs in an automated way the random sampling, multiple code execution, and data processing to derive the tolerance limits. For an uncertainty analysis with RELAP/SCDAPSIM three phases are required (Perez Ferragut 2012), as follows:

- The "setup" phase generates the total number of sampled values, also called "weights", and information needed to build the tolerance bounds during the "post-processing" phase. The weights are used to associate uncertainty to code parameters by applying them as multipliers to the base values. During this phase the code also computes the required number of code runs by using the Wilks' formula, or simply uses the value supplied by the user;

- The "simulation" phase consists of the base case run in which the simulation is done as if there were no uncertainty option available, and the set of uncertainty runs which have input and source modifications. Except for the base case run, each run of the simulation phase reads its corresponding weight file generated by the "setup" phase for that run;

- The "post-processing" phase reads the restart-plot files written during the base case and the uncertainty runs and generates the rank matrices for the output quantities defined in the "post-processing" input file. The rank

matrices contain the values for the output parameters sorted according to its rank and are used to determine the tolerance intervals. The information required in the "post-processing" input file also includes the simulation runs to be used in the generation of the tolerance intervals.

CANDU 6 MODEL IN RELAP/SCDAPSIM FOR A LB-LOCA ACCIDENT

Full plant model

The full plant model (as shown in Figure 1) used in this analysis is a simple model of the core, containing 4 fuel channels (95 fuel channels lumped into a single channel, the core power being equally distributed to each channel); the fuel assemblies along with the pressure tube were modelled using RELAP heat structures. The calandria vessel, calandria tubes and annular gas between the calandria tubes and pressure tubes were not modelled.

Each fuel channel was divided into 12 axial volumes corresponding to the 12 fuel bundles inside it.

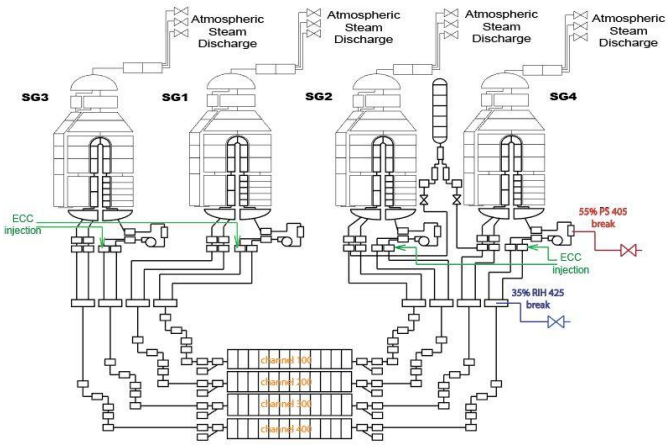


Figure 1: CANDU 6 LB-LOCA model

The RELAP5 thermal-hydraulic model solves eight field equations for eight primary dependent variables. The primary dependent variables are pressure (P), phasic specific internal energies (U_g, U_f), vapor volume fraction (void fraction) (α_g), phasic velocities (v_g, v_f), noncondensable quality (X_n), and boron density (ρ_b). The independent variables are time (t) and distance (x). The secondary dependent variables used in the equations are phasic densities (ρ_g, ρ_f), phasic temperatures (T_g, T_f), saturation temperature (T_s), and noncondensable mass fraction in noncondensable gas phase (X_{ni}). The equations are described in the Models and Correlations Code Manual (NUREG/CR-5535/Rev 1-Vol IV)

The present model was developed at Politehnica University of Bucharest more than 10 years ago (Prisecaru et al. 2005), with slight variations up to the present, regarding the number of the thermal hydraulic fuel channels describing the core (4 to 16 fuel channels), fuel bundles modelling (using RELAP heat structures or SCDAP core components) in order to better represent the CANDU 6 behaviour under different accident condition.

Fuel bundle model

The fuel bundle of a CANDU 6 reactor consists of 37 elements displaced on 3 concentric rings as shown in Figure 2. Inside the pressure tube 12 fuel bundles are located, the axial power distribution following a cosine shape, with the maximum in the centre of the channel (bundles 6 and 7), and decreasing to the first and last fuel bundle (bundles 1 and 12 producing less power than the central ones).

Figure 2: CANDU 6 fuel bundle (core displacement)

The heat structures in RELAP5 permit the calculation of heat across the solid boundaries of the hydrodynamic volumes. Heat transfer can be modeled from and/or through structures, including fuel pins, steam generator tubes, and pipe and vessel walls. One-dimensional heat conduction in cylindrical geometry can be represented by the heat structures in RELAP5. Surface multipliers are used to convert the unit surface of the one-dimensional calculation to the actual surface of the heat structure. Thermal conductivities and volumetric heat capacities as functions of temperature are described using tables in the input file.

Finite differences are used to advance the heat conduction solutions. Each mesh interval contain a mesh spacing, and a different material.

Symmetrical or insulated boundary conditions can also be simulated. For heat structure surfaces connected to hydrodynamic volumes, a heat transfer package containing correlations for convective, nucleate boiling, transition boiling, and film heat transfer from the wall-to-water and reverse transfer from water-to-wall is provided.

It is assumed in one-dimensional heat conduction that the temperature distribution in the axial or radial direction is the same throughout the structure being modeled and that the linear heat flow is negligible. The equation governing one dimensional heat conduction for a cylindrical geometry is:

$$\rho C_p \frac{\partial T}{\partial t} = \frac{1}{r}\left[\frac{\partial}{\partial r}\left(rk\frac{\partial T}{\partial r}\right)\right] + S$$

where T is the temperature, t is the time, r is the radius, S is the internal heat source, ρCp is the volumetric heat capacity, and k is the thermal conductivity.

In order to model a heat structure in RELAP5, a mesh is set up beginning at the left boundary of the structure being modeled and continuing to the right boundary. The mesh point spacing (Figure 3) is taken as positive as x or r increases from left to right. Mesh points must be placed on the external boundaries of the structure unless a symmetrical or adiabatic boundary condition is to be used. Mesh points may also be placed at any desired intervals within the structure and should be placed at the interfaces between the different materials. The spacing of the mesh points may vary from material to material and may vary within the material as the user desires.

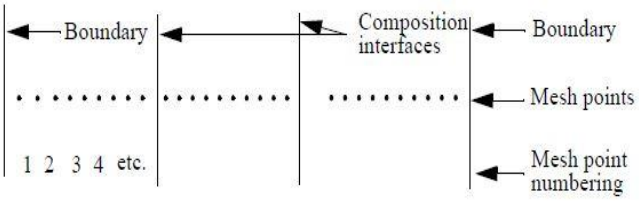


Figure 3: Mesh point layout

The RELAP5 gap conductance model accounts for the first-order effects of material deformations under normal reactor operating conditions and most postulated LOCA conditions. The model considers, among other things, the thermal expansion of the fuel and the cladding, and the elastic deformation of cladding under the differential pressure between the gas internal to the gap and the fluid outside the cladding.

The 37 fuel pins of the fuel bundle are combined into a single fuel pin heat structure maintaining the surface area, mass and equivalent heated perimeter. The fuel pin is radially discretized in 5 regions/intervals simulating the different layers (3 intervals for the fuel, 1 interval for the gap, and 1 interval for the cladding) as shown in Figure 4.

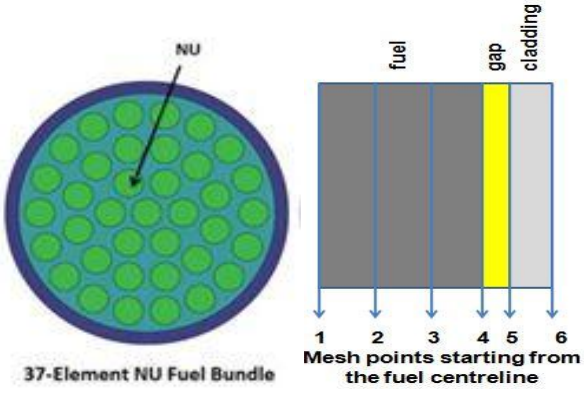


Figure 4: RELAP5 heat structure for the fuel bundle

Break model

The break is modelled using a connecting trip valve which opens at the specified time from the beginning of the analysis.

Valves are quasi-steady models that are used either to specify an option in a system model or to simulate control mechanisms in a hydrodynamic system. The operation of a trip valve is solely dependent on the trip selected. With an appropriate trip, an abrupt full opening or full closing of the valve will occur. A latch option is also included for latching the valve in the open or closed position. The model for this type of valve does not include valve inertia or momentum effects. If the valve is used as a junction with an abrupt area change, then the abrupt area change model is used to calculate kinetic loss factors when the valve is open.

Modelling assumptions

The main assumptions used in this study are:

- All heat transport pumps are assumed available until tripped by protective system;
- Pressurizer is connected prior loop isolation.
- The D_2O feed and bleed system is modeled and is assumed to be available according to normal inventory control logic;
- Emergency core coolant injection is available;
- Feedwater control valve is available in order to control the steam generator level.

UNCERTAINTY ANALYSIS

In the setup phase of the uncertainty analysis, the input file is prepared, according to the specifications from the manual (Perez-Ferragut 2015). The code uses the Wilks' formula to compute the required number of uncertainty runs given the:
- Percentile (<1) or coverage of the tolerance limit.
- Confidence level (<1).
- Order for Wilks' formula application (1, 2 . . .).

For this paper Wilks` formula of the 2nd order was used to derive the estimations of the 5th and 95th percentile with a 0.95 confidence level of the maximum cladding temperature reached during the break.

A total of 30 parameters (28 parameters were considered before in a previouse analysis performed by Dupleac et. al 2011, with two additional parameters presented in Table 1) were selected for the uncertainty analysis. They cover initial and boundary conditions, pressure drops, material properties and heat transfer correlations including critical heat flux.

The table providing the phenomena, the associated code parameter, the Probabilistic Distribution Function (PDF) type to describe the uncertainty and additional comments to clarify how the uncertainty was applied. The intial parameters were classified as “input treatable parameters”

and "source correlation parameters" depending upon their modification can be done from the regular input file or need for a source modification. The information required by the code for the first type is the input card and word number holding the base case value of the parameter, while the latter type requires the specification of a code given name that identifies the correlation coefficient to be modified. The uncertainty of the parameters was determined according to code documentation (NUREG/CR-5535/Rev 1-Vol IV), phase 5 of BEMUSE programme (Reventos et al.), plant instrumentation information, and expert judgment when no related information could be found.

Table 1: Additional uncertainty parameters

INPUT TREATABLE PARAMETERS (additional)				
#	Phenomena	Parameter	Distribution	Comments
1	LOCA signal	Pressure in the RIH425	ND	Multiplier applied to the pressure signal
2	ECC injection	Accumulator water volume	UD	Multiplier applied to the water volume

According to Wilks' formula, the execution phase consisted in 93 code runs. None of the code runs failed, so all the 93 code runs could be included in the uncertainty evaluation of the maximum cladding temperature.

According to order statistics theory, the 95/95 unilateral tolerance limit is given by rank number 92, *i.e.* the second largest value, and covers the 95th percentile of the output quantity with a confidence level of 0.95. On the other hand the 5/95 unilateral tolerance limit is given by rank number 2, *i.e.* the second smallest value, and covers the 5th percentile of the output quantity with a confidence level of 0.95.

LB-LOCA ANALYSIS AND RESULTS

A LB LOCA involves a break in the PHTS boundary of sufficient magnitude that the Reactor Regulating System (RRS) is incapable of maintaining reactivity balance. As the pressure tubes and feeder pipes are of relatively small diameter this type of LOCA can only happen due to a break in the larger headers above the reactor core or at the pump suction. Two break locations were considered in the analysis:

- Case A: 35% Reactor Inlet Header (RIH) break
- Case B: 55% Pump Suction (PS) break

The initial phase of the accident (0-5 s) is characterised by a short power transient, which ends by either a neutron or process trip. The main safety concern for this short period prior to reactor trip is that the fuel sheath and UO_2 temperature could rise and turn into molten material, which could potentially cause pressure tube rupture. In turn the resulting hot spots in the pressure tube could result in localised straining and possible failure of the pressure tube (NEA OECD 2009). This safety concern is not met as the fuel is cooled by the flow of coolant resulting from the blowdown.

The second phase of the accident (5-30 s) is characterised by the blowdown and depressurization of the fuel channel prior to ECC injection. Despite the reactor shutdown, fuel temperatures may remain high due to the cooling degradation, decay heat and oxidation of the fuel sheath. The fuel sheath may undergo significant deformation and may fail releasing fission products to the fuel channel and subsequently to containment.

The third phase of the accident (30-200 s) is characterised by the initiation of ECC. During this period, ECC is being injected into the primary heat transport system, but has not yet reached sufficient levels to effectively cool the fuel. Depressurisation of the heat transport system continues and stored heat and decay heat from the fuel is radially removed to the moderator through the pressure tube and calandria tube. Fuel failures are likely during this stage of the accident.

During the fourth and final phase of the accident (>200 s) the injection of ECC has reached a level where it can effectively cool the fuel. The heat transport system pumps have tripped, refill of the channels in the core proceeds and a quasi-steady state is attained. This phase was not considered in the analysis.

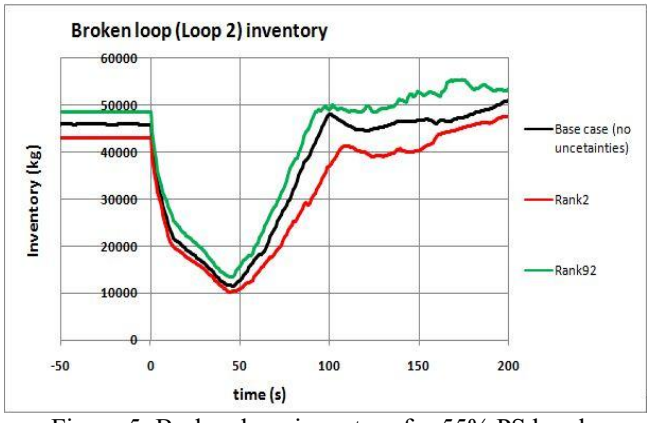


Figure 5: Broken loop inventory for 55% PS break

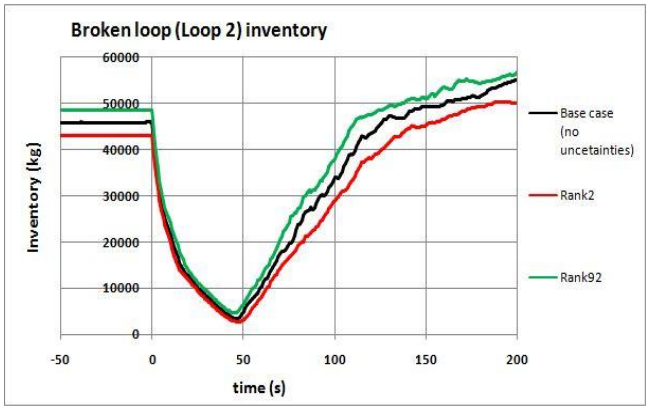


Figure 6: Broken loop inventory for 35% RIH break

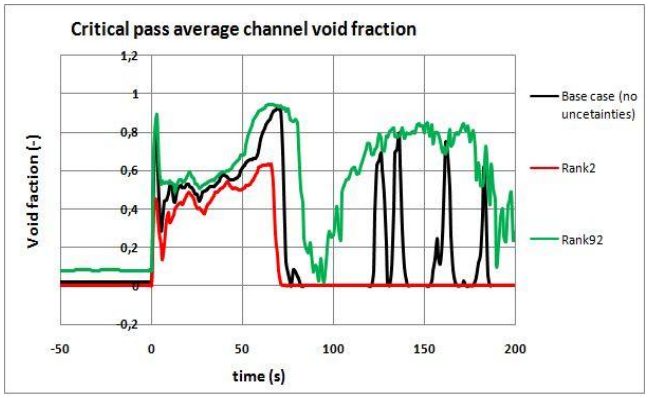


Figure 7: Void fraction pass 4 at 55% PS break

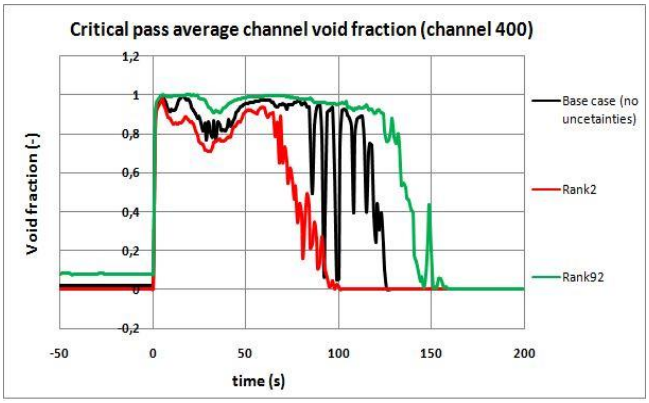


Figure 8: Void fraction pass 4 at 35% RIH break

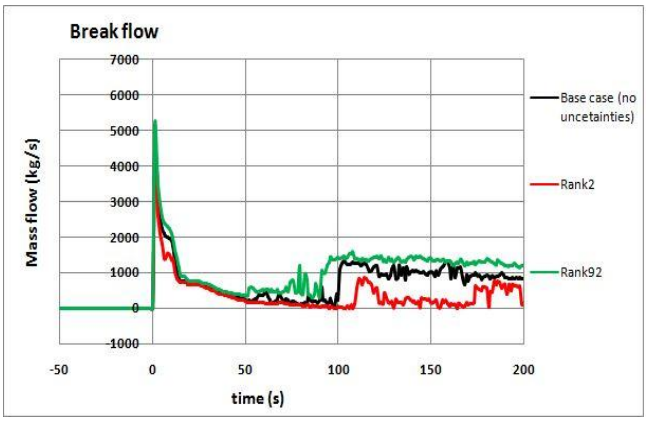


Figure 9: Mass flow at the 55% PS break

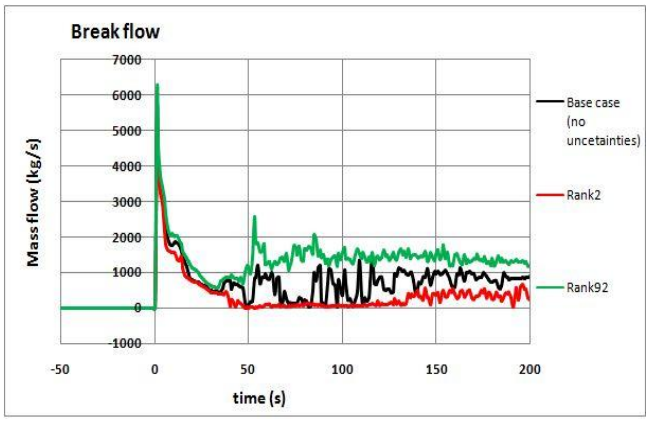


Figure 10: Mass flow at the 35% RIH break

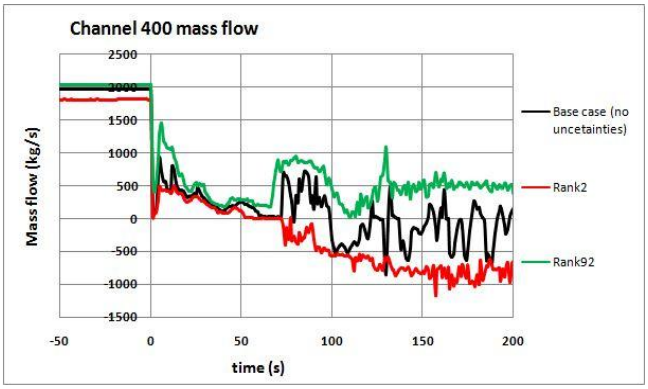


Figure 11: Mass flow in channel 400 at 55% PS break

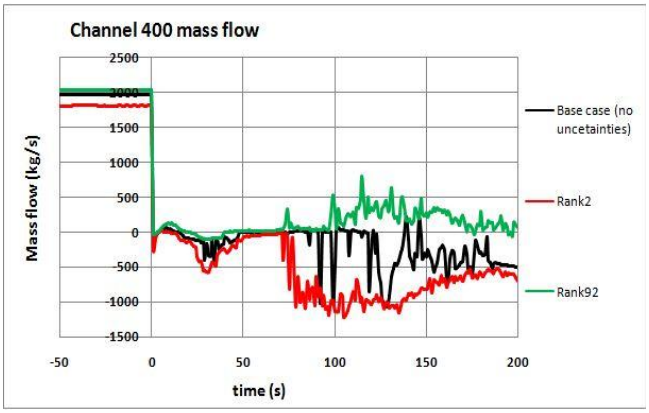


Figure 12: Mass flow in channel 400 at 35% RIH break

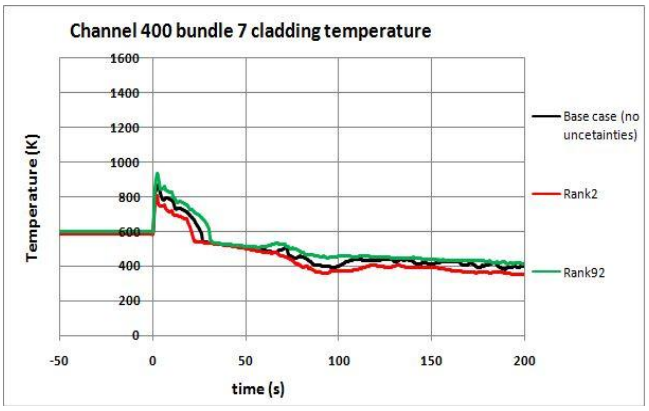


Figure 13: Sheath temperature at 55% PS break

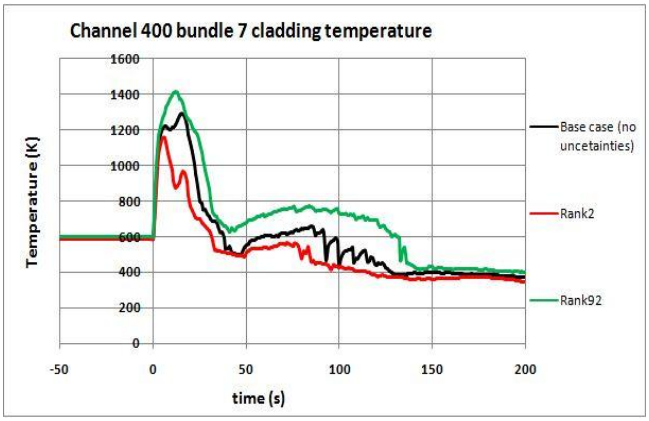


Figure 14: Sheath at 35% RIH break

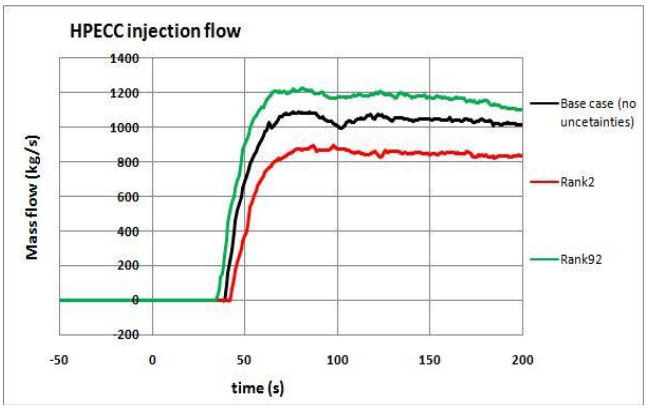


Figure 15: ECC flow at 55% PS break

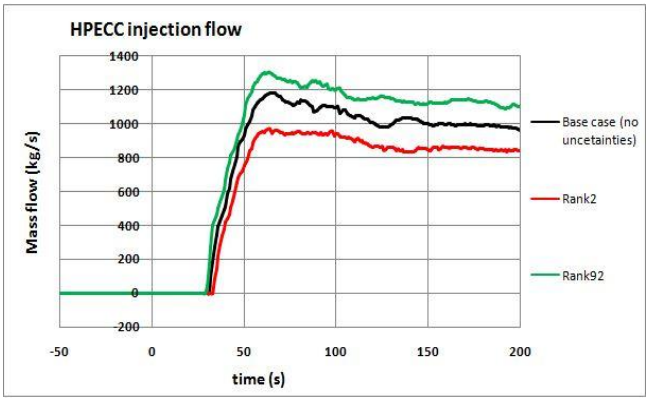


Figure 16: ECC flow at 35% RIH break

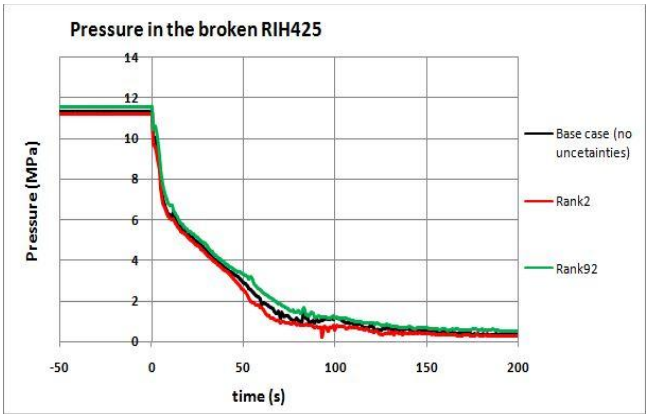


Figure 17: RIH 425 pressure 55% PS break

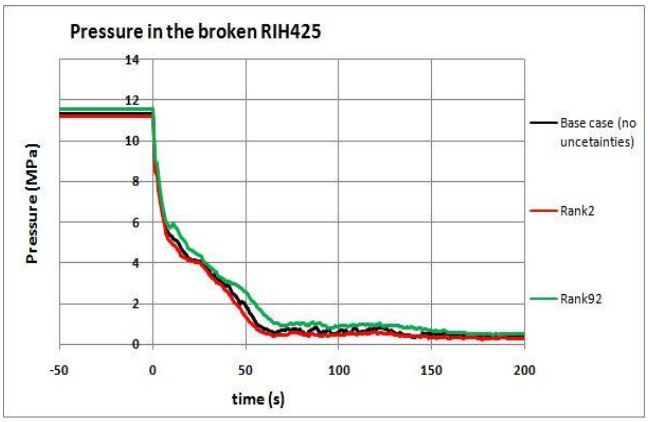


Figure 18: RIH 425 pressure 35% RIH break

CONCLUSIONS

This analysis was performed in order to determine the maximum cladding temperature when two different break location were considered when uncertainties were accounted. The break sizes are selected based upon the prediction of the highest temperatures of the cladding in the previous studies.

The -50-0s period of the analysis is showing the steady state analysis for the balance of the plant. The initiating event occurs at 0s and consequently the loop inventory decreases rapidly (as is shown in Figure 5 and Figure 6) up to the point where the ECC starts to inject water in the PHTS due to a pressure signal, right after this moment the loop inventory starts to recover.

The void fraction of the critical pass never reaches to 1 for the 55% PS break (Figure 7), while for 35% RIH break reaches the maximum shortly after the initiating event (Figure 8) and decreases with the ECC injection.

Due to the high pressure in the PHTS, once the valve modelling the break opens, a very large amount of coolant is expulsed to the containment, out of the PHTS boundary, but it decreases rapidly due to the depressurization of the PHTS (as shown in Figures 17 and 18). After the ECC injection reaches the maximum flow, the flow at the break decreases significantly.

The peak cladding temperature for the most affected channel (channel 400) reaches 1400K for the 35% RIH break, while for the 55% break does not go beyond 1000K. The uncertainty analysis showed a 97K difference between the base case and the upper limit 95/95, described by rank 92, which occurs at 12s after the initiating event. Is noticeable that the uncertainty band is larger for the 35% RIH break considered.

When the pressure in the PHTS falls below the ECC injection pressure, ECC injection begins in the broken loop (Figure 15 and Figure 16). The uncertainty analysis shows that the start moment of the injection could slightly vary for the 55% PS break compared to the 35% RIH break. The mass flow injected in both cases showed similar trends.

REFERENCES

Dupleac, D., Perez, M., Reventos, F., Allison, C., 2011. "Uncertainty analysis of the 35% reactor inlet header break in a CANDU 6 reactor using RELAP/SCDAPSIM/MOD4.0 with integrated uncertainty analysis option." *Proceedings of The 14th International Topical Meeting on Nuclear Reactor Thermalhydraulics*, NURETH-14, Toronto, Ontario, Canada

Perez Ferragut, M. 2012. "Integration of a quantitative-based selection procedure in an uncertainty analysis methodology for NPP safety analysis." PhD thesis, Barcelona, Spain

Prisecaru, I., Dupleac, D., Biro, L., 2005. "A study of RELAP5 capability to perform the analysis of CANDU 6 reactors

accidents.", *Proceedings of The 14th International Topical Meeting on Nuclear Reactor Thermalhydraulics*, NURETH-14, Avignon, France

Reventos, F., Batet, L., Perez, M., "BEMUSE phase V report: Uncertainty and sensitivity analysis of a LB-LOCA in ZION nuclear power plant" NEA/CSNI/R(2009).

NEA OECD 2009. "Nuclear Fuel Behaviour in Loss-of-coolant Accident (LOCA) Conditions." Nuclear Energy Agency. Organisation for Economic Co-operation and Development. ISBN 978-92-64-99091-3

Nuclear Safety Analysis Division, 2001. "RELAP5/MOD3.3 CODE MANUAL VOLUME IV. Models and Correlations.", NUREG/CR-5535/Rev 1-Vol IV. Idaho Falls, Idaho, December 2001

AUTHOR BIOGRAPHY

ROXANA-MIHAELA NISTOR-VLAD is a PhD student, with the main focus of the thesis on the Best Estimate Plus Uncertainty analysis of the CANDU 6 reactors. Graduated the Politehnica University of Bucharest in 2013, getting the Bachelor`s degree in Power Engineering and Nuclear Technologies, and the Master`s degree in 2015 in Nuclear Engineering. During the master studies she worked for CITON in the Nuclear Safety Department as a junior engineer, one of the main activities being the development of detailed input models and user guidelines for CANDU reactors using RELAP/SCDAPSIM code. Also, an advanced user of the code after getting a full training at Innovative Systems Software (USA), and working to develop the full plant display using GRAPE (Graphical RELAP/SCDAPSIM Analysis Platform for Education and Engineering).

ELECTRONIC NETWORK SIMULATION

NEW SCHEDULING MECHANISM IN MULTI-CHANNEL RECONFIGURABLE WSN UNDER QOS AND ENERGY CONSTRAINTS

Yousra Ben Aissa
Faculty of Mathematical, Physical and Natural Sciences, University of Tunis-El Manar, Tunisia.
Email: yousra.benaissa@fst.utm.tn

Olfa Mosbahi
Mohamed Khalgui
National Institute of Applied Sciences and Technology
University of Carthage, Tunisia.
Email: olfamosbahi@gmail.com
Email: khalgui.mohamed@gmail.com

Abdelmalik Bachir
LESIA Lab, Computer Science Department,
Biskra University, Algeria.
Email: a.bachir@univ-biskra.dz

KEYWORDS
Reconfigurable wireless sensor network, Real-time message, Multi-channel communication, Energy harvesting.

ABSTRACT

This paper deals with multi-channel reconfigurable WSN (RWSN) under time and energy constraints, that adapts its behavior at run-time to any evolution in the related environment according to well-defined conditions. A reconfiguration scenario is adding new tasks and consequently new messages on the medium, removing other ones or changing their characteristics at run-time, which can lead the RWSN to violate time and energy constraints. This highlights the need for a scheduling mechanism of exchanged traffic on multi-channel RWSN for guaranteeing the system feasibility under real-time and energy constraints. The proposed mechanism RController achieves significant improvements over existing methods and provides the highest percentage of adding messages that can go up to 85% while meeting deadlines, with a lower average in response time and energy consumption.

INTRODUCTION

A wireless sensor network (WSN) consists of three main components: sensor nodes, gateways, and software, where the main issues in sensor nodes are the limited source energy which is a battery, and the sensed data can be periodic and have deadlines Abdul-Salaam et al. (2016). To overcome these limitations, we need to provide a mechanism to extend sensor nodes life-time. Energy harvesting is a mechanism used to generate electrical energy from the unused ambient energy, where it is ideal for applications that need to survive for a longer time periods . Its techniques include photovoltaic which is the most appropriate in a lighted environments Ferdous et al. (2016). Integrating real-time system and wireless communication is a non-trivial task, especially for environmental monitoring applications that require network performance in terms of reliability and deliver timely messages, which have strict constrains on the quality of service (QoS) and energy consumption, where the messages response time is one of the major QoS parameters, and these constraints are controlled in a finite time Respondek (2010). Decent to these strict constraints, WSN needs to adapt its behavior to the environment according to circumstances at run-time Zhang et al. (2013). Accurately, a WSN is called reconfigurable (RWSN) when any modification touches its sensor nodes. In this paper, we are concerned with real-time message transmission on multi-channel, where for messages transmission the system needs to meet real-time and energy constraints. This highlights the need for an RWSN scheduling mechanism under system feasibility in time and energy. The scheduling in multi-channel RWSN can be divided into two issues: The first is like a bin packing problem, where messages with different types must be sent over a finite number of channels, in the way to maximize the messages percentage and minimize the used number of channels. The second issue deals with the scheduling of real-time messages on one channel. By nature, the system is feasible in time and energy, but after applying reconfiguration scenarios that add new tasks and messages, the system can become non-feasible in time or in energy. Thus, we propose a new scheduling mechanism for multi-channel RWSN (RController) that maintains system feasibility while adding messages. RController uses linear programming for solving this issue. The contribution is applied to a case study then compared with the existing methods to present its originality. RController achieves significant improvement over existing methods and provides the highest percentage of adding messages under time and energy constraints that can go up to 85%, with a lower average in response time and energy consumption. The main original contribution of this paper is to: i) determine the path that will be used to send messages, ii) compute the set of channels needed to be active, iii) find the set of periodic tasks and messages that can be feasible, iv) divide message WCTTs on multi-channels. This paper is organized as follows: The first section summarizes the existing related work. The second sec-

tion present the formal model. In the third section, we present the proposed contribution. The fourth section evaluates the performance of RController. Finally section provides concluding remarks andfuture work.

RELATED WORK

Most of the recent research has been carried out guarantee QoS of WSN under real-time and energy constraints. The work reported in Chipara et al. (2013), the authors propose a real-time query scheduling (RTQS). Then as reported in Nasser et al. (2013), the authors propose a dynamic multi-level priority (DMP) for packet scheduling simultaneously in WSN according to the first come first served (FCFS). The work reported in Ghribi et al. (2018) proposes an energy aware real-time scheduling algorithm. The work reported in Housseyni et al. (2018) develops a multi-agent architecture for the real-time scheduling of distributed reconfigurable embedded systems, which is called EDH. The authors in Wu et al. (2016) assign the senders and corresponding receivers to achieve the delay and energy constraints. To the best of our knowledge no one has considered before real-time and energy assumptions before sending messages. No one has proposed a solution to make the non-feasible system feasible in time and energy. Thus, the proposed scheduling mechanism RController in this context are totally original in multi-channel RWSN.

FORMALIZATION MULTI-CHANNEL RWSN

In this section, we give a mathematical representation of an RWSN under time and energy constraints.

System Overview

In this paper, we consider an RWSN composed of reconfigurable nodes powered by batteries and capable of performing energy harvesting. We denote by $\mathcal{M}$ the set of messages that are planned to be transmitted, and by *Paths* the set of paths that will be used to send messages from a source node to a sink node. We assume that each path $\mathcal{P} \in Paths$ is a set of mediums (or radio interfaces) that allow simultaneous parallel transmissions over a set of channels $\mathcal{C}$. We denote by $C_B(t)$ the residual battery capacity at time t, P_{tx} (resp. P_{rx} and P_{idle}) the power consumption in transmission (resp. in reception and idle mode), T_{tx} (resp. T_{rx} and T_{idle}) the transmission (resp. reception and idle) time, $\mathcal{M}_s^j$ the messages that will be transmitted over channel c_j, $m_i \in \mathcal{M}$,$i = 1..|\mathcal{M}|$ the new message which will be added to the system, r_i the arrival time, T_i the period, D_i the deadline, $WCTT_{i,j}$ the worst case transmission time, $WCEC_{i,j}$ the worst case energy consumption.

Time Feasibility

Checking system feasibility is verified by considering the scheduling of real-time messages that is done according to the earliest deadline first (EDF). Thus, the total channel utilization U_j is not more than 100% . The system is time-feasible when sending messages $\mathcal{M}_s^j$ over channel c_j if they are sent without exceeding their deadlines at each period. Thus, the total channel utilization with EDF is given by Stankovic et al. (2012)

$$U_j = \sum_{i=1}^{|\mathcal{M}_s^j|} \frac{WCTT_{i,j}}{T_i} \leq 1 \tag{1}$$

Energy Feasibility

Energy Harvesting Model
Sensor node is able to work on energy harvesting which is collected from a photovoltaic (PV) power source. The output power generated from PV at time t referred to as $P_{PV}(t)$ is given in Lee (2011), and by assuming that the energy is harvested uniformly over a time interval, the amount of energy harvesting $E_H([t_1, t_2])$ in time interval $[t_1, t_2]$ can be given by

$$E_H([t_1, t_2]) = \int_{t=t_1}^{t_2} P_{PV}(t)dt \tag{2}$$

Energy Consumption Model
In this section, we focus on the consumed energy during communication, which is equal to the sum of the consumed energy by the radio of a node in the following modes: transmission, reception, and idle as reported in Kurt et al. (2017). Therefore, the consumed energy in communication by a channel c_j over a period of time $[t_1, t_2]$ is given by the following formula 3, where $\triangle t$ is the length of $[t_1, t_2]$.

$$E_j([t_1, t_2]) = \sum_{i=1}^{|\mathcal{M}_s^j|} (P_{tx}WCTT_{i,j} + P_{rx}T_{rx} + P_{idle}T_{idle}) \frac{\triangle t}{T_i} \tag{3}$$

As consequence, the system is feasible in energy in the interval $[t_1, t_2]$ if

$$\sum_{j=1}^{|\mathcal{C}|} E_j([t_1, t_2]) < C_B(t_1) + E_H([t_1, t_2]) \tag{4}$$

RController: SCHEDULING IN MULTI-CHANNEL RWSN

The arrival of new messages can make the system non-feasible in time or energy, thus we propose a new scheduling mechanism for multi-channel RWSN called RController (reconfigurable controller), for maintaining system feasibility while adding messages over the whole path. The main idea behind RController is to make the system feasible in time and energy allowing all messages to meet their deadline, by selecting messages which will

be added on the medium and determining if each message will be sent on one channel, will be divided its WCTT and send it on multiple channels, or will be ignored. This procedure is done in each period of time, and it is called reconfiguration time period. When nodes sense an event, and before sending the information message, they send an alert to the sink node by single or multi-hop routing. This alert contains the size and period of the message. If the sink node receives these alerts in the reconfiguration time period, then it will call RController to discover which paths will be used by the source nodes. These paths are chosen once the system feasibility in time and energy is verified. These chosen paths will be broadcasted as a reconfigurable matrix. then messages will be scheduled according to EDF algorithm. Otherwise, if the alerts have arrived outside the reconfiguration time period, then they will be processed later due to security reasons. RController is applied by using linear programming, where each message m_i has a coefficient $\alpha_{i,j}$ (rounded to 10^{-x}, where x is defined based on m_i) which mention that this message will be sent over channel c_j or not, and according to which method it will be sent, as following

$$\begin{cases} \text{Maximize} & \sum_{i=1}^{|\mathcal{M}|}\sum_{j=1}^{|\mathcal{C}|}\alpha_{i,j} & (8.1) \\ \text{Subject to} & \exists \mathcal{P} \in \text{Paths},\ \forall \mathcal{C} \in \mathcal{P}, \forall \text{c}_\text{j} \in \mathcal{C}: & \\ & U_j = \sum_{i=1}^{|\mathcal{M}|} \frac{\alpha_{i,j} WCTT_{i,j}}{T_i} \leq 1 & (8.2) \\ & E_j([t_1,t_2]) < C_B(t_1) + E_H([t_1,t_2]) & (8.3) \\ & \forall m_i \in \mathcal{M}_s^{\mathcal{C}},\ \sum_{j=1}^{|\mathcal{C}|}\alpha_{i,j} = 1 & (8.4) \\ & \forall m_i \in \mathcal{M},\ 0 \leq \alpha_{i,j} \leq 1 & (8.5) \end{cases} \quad (5)$$

Where, the objective function is to minimize channels number and to maximize the number of messages added over them as coming in Eq.8.1. Minimizing channels number means maximizing the sum of $\alpha_{i,j}$ over channels set $\mathcal{C}$, then maximizing messages number means maximizing the sum of $\alpha_{i,j}$ for each message $m_i \in \mathcal{M}$. Thus, the objective function is the sum of the sum of $\alpha_{i,j}$. Also it is subject to linear constraints: i) system feasibility must be maintained in time (see Eq.8.2), where RController has to choose the value of $\alpha_{i,j}$ which allows to add message m_i over channel c_j, ii) system feasibility must be maintained in energy as coming in Eq.8.3, where the consumed energy by each channel must satisfy this equation, iii) for each added message m_i the sum of $\alpha_{i,j}$ over the set of channels C must equals to 1 (see Eq.8.4), which means that m_i will be sent over these channels according to $\alpha_{i,j}$, iv) for each $\alpha_{i,j}$ its value must be between one and zero as coming in Eq.8.5, where each value has a signification: if the value of $\alpha_{i,j}$ is equal to one, then message m_i will be sent over one channel which is c_j. Else, if it is between zero and one, then the message will be sent over multiple channels, where the WCTT of this message will be multiplied by $\alpha_{i,j}$ for each channel c_j (i.e., $\alpha_{i,j}WCTT_{i,j}$). Otherwise, the message m_i will be ignored, as following

$$\forall m_i \in \mathcal{M}, \forall c_j \in \mathcal{C}, \alpha_{i,j} \begin{cases} 1 & \text{one channel} \\ 0 < \alpha_{i,j} < 1 & \text{multi-channel} \\ else & \text{ignored} \end{cases} \quad (6)$$

EXPERIMENTATION

Case Study

We consider an RWSN where its system is initially feasible, then after adding a set of messages (see Table 1), the system will be non-feasible. Thus we run a set of reconfiguration scenarios to add messages and maintain system feasibility, as follows

Table 1: An example for the addition of a set of messages.

m_i	r_i	$WCTT_i$	T_i	m_i	r_i	$WCTT_i$	T_i
0	0	70	300	1	0	80	100
2	0	40	200	3	0	60	600
4	0	50	350	5	0	40	550
6	0	60	100	7	0	50	200
8	0	80	200	9	0	70	300
10	0	80	100	11	0	40	200
12	0	60	600	13	0	50	350
14	0	40	550	15	0	60	100
16	0	50	200	17	0	80	200
18	0	70	300	19	0	80	100
20	0	40	200	21	0	60	600
22	0	50	350	23	0	40	550
24	0	60	100	25	0	50	200
26	0	80	200	27	0	70	300
28	0	80	100	29	0	40	200
30	0	60	600	31	0	50	350
32	0	40	550	33	0	60	100
34	0	50	200	35	0	80	200

The sequence of reconfiguration scenarios and application of the proposed model in the medium 1:
- *Initial messages in:* $channel_1$: $\mathcal{M}^1 = \{18, 23, 27, 32\}$, $channel_2$: $\mathcal{M}^2 = \{\}$, $channel_3$: $\mathcal{M}^3 = \{\}$
- *Initial channel utilization:* $U_1 = 0.612, U_2 = 0, U_3 = 0$
- *After using RController, the messages set in:* $channel_1$: $\mathcal{M}^1 = \{14, 18, 22, 23, 27, 31, 32\}$, $channel_2$: $\mathcal{M}^2 = \{5, 7, 9, 12, 13, 25\}$, $channel_3$: $\mathcal{M}^3 = \{0, 2, 3, 4, 11, 25\}$
- *Channel utilization:* $U_1 = 0.97, U_2 = 1, U_3 = 0.93$
- *Interpretations:* The system is feasible in time and in energy for sending $\{m_{14}, m_{22}, m_{31}\}$ over $Channel_1$, so they are added without any reconfiguration. Then for $\{m_0, m_2, m_3, m_4, m_5, m_7, m_9, m_{11}, m_{12}, m_{13}\}$ the system will be feasible after enabling new channels $Channel_2$ and $Channel_3$ for sending these messages. Although, the system is not feasible in time or in en-

ergy for sending the other messages, message m_{25} will be sent over both channels $Channel_2$ and $Channel_3$ by dividing its $WCTT$, where 80% of its $WCTT$ will be passed over $Channel_2$ and 20% over $Channel_3$.
The sequence of reconfiguration scenarios and application of the proposed model in the medium 2:
- *Initial messages in:* $channel_1$: $\mathcal{M}^4 = \{\}$, $channel_2$: $\mathcal{M}^5 = \{\}$, $channel_3$: $\mathcal{M}^6 = \{18, 21, 32\}$
- *Initial channel utilization:* $U_4 = 0$, $U_5 = 0$, $U_6 = 0.406$
- *After using RController, the messages set in:* $channel_1$: $\mathcal{M}^4 = \{5, 7, 9, 12, 13, 14, 25\}$, $channel_2$: $\mathcal{M}^5 = \{0, 2, 3, 4, 11, 25\}$, $channel_3$: $\mathcal{M}^6 = \{18, 21, 22, 23, 27, 31, 32\}$
- *Channel utilization:* $U_4 = 1$, $U_5 = 0.999$, $U_6 = 0.998$
- *Interpretations:* The system is feasible in time and energy for sending $\{m_{22}, m_{23}, m_{27}, m_{31}\}$ over $Channel_6$, so they are added without any reconfiguration. Then for the other messages, there are which can be added after enabling channels $Channel_4$ and $Channel_5$. Although, message m_{25} can be sent over both channels $Channel_4$ and $Channel_5$ by dividing its $WCTT$, where 51% of its $WCTT$ will be passed over $Channel_4$ and 49% over $Channel_5$.
The sequence of reconfiguration scenarios and application of the proposed model in the medium 3:
- *Initial messages in:* $channel_1$: $\mathcal{M}^7 = \{16\}$, $\mathcal{M}^8 = \{\}$, $\mathcal{M}^9 = \{34\}$
- *Initial channel utilization:* $U_7 = 0.25$, $U_8 = 0$, $U_9 = 0.25$
- *After using RController, the messages set in:* $channel_1$: $\mathcal{M}^7 = \{5, 9, 12, 14, 16, 25\}$, $channel_2$: $\mathcal{M}^8 = \{0, 2, 3, 4, 7\}$, $channel_3$: $\mathcal{M}^9 = \{11, 13, 22, 31, 34\}$
- *Channel utilization:* $U_7 = 0.979$, $U_8 = 0.926$, $U_9 = 0.879$
- *Interpretations:* The system is feasible in time and energy for sending $\{m_{11}, m_{13}, m_{22}, m_{31}, m_{25}\}$ over $Channel_9$ and $\{m_5, m_9, m_{12}, m_{14}\}$ over $Channel_7$, then after enabling $Channel_8$ the system will be feasible to send $\{m_0, m_2, m_3, m_4, m_7\}$ over it, and the other messages will be ignored due to system feasibility.
In this simple case study the percentage of added messages on a path with three mediums is 58.33%.

Simulation and Evaluation

In order to evaluate the impact of RController on deadline success, we measure the percentage of successfully added messages, i.e., those messages that have been added while ensuring that the system is feasible in time and energy. For the simulation, we use 300 random messages. Fig. 1 shows the added messages and the corresponding channels number used to send these messages on each medium after using RController (we plot only the first fifteen messages in this figure). As shown in Fig.1, the messages $\{m_1, m_3, m_9, m_{10}, m_{12}, m_{13}, m_{14}\}$ will be sent over one channel in each medium in a path, and the message m_5 will be sent over one channel for the first three mediums in the path, then it will be divided into two channels for the last three mediums in the same path, the same for the message m_7, where it will be divided into three channels for the first three mediums, then it will be sent over two channels for the next three mediums in the same path. The remaining messages will be ignored. To evaluate the effect of varying the number of available channels on the performance of the considered strategy, the Fig. 2 shows that RController achieves the highest addition success rate than EDH and DMP, where the percentage of messages added is up to 85% when we use six channels compared to EDH and DMP which is between 30% and 40%. To evaluate the effect of varying the number of channels on the response time, Fig.3 shows that RController provides the lowest average response time than EDH when the number of used channels between two and five, and DMP provides the highest average in response time than both RController and EDH. To evaluate the effect of varying channels number on the consumed energy of the considered strategies, Fig. 4 shows that RController achieves the lowest average in consumed energy than EDH and DMP when the number of used channels is between two and four. These experiments show that significant improvements can be obtained when the RWSN uses RController on multi-channel compared with the

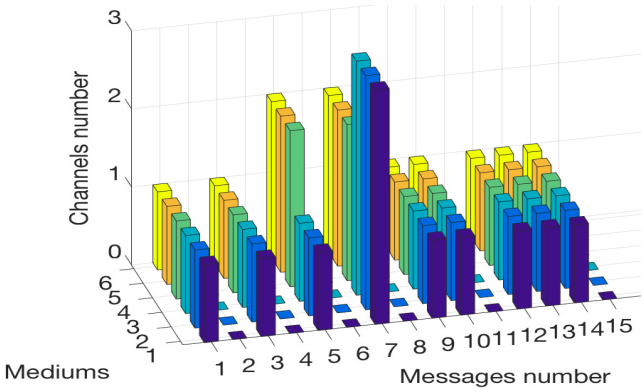


Figure 1: Number of channels used by each message on each medium.

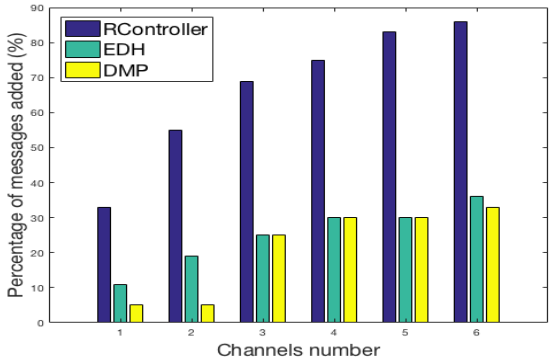


Figure 2: Percentage of messages sent based on channels number.

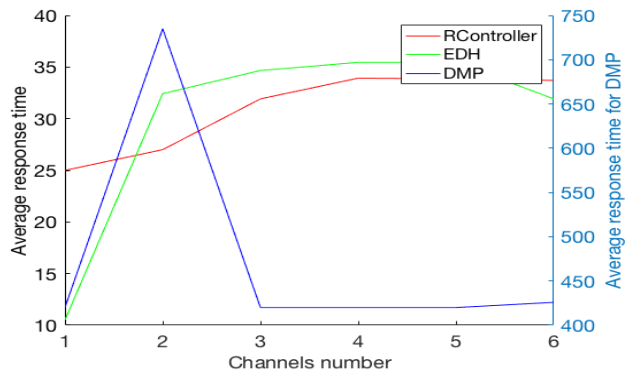


Figure 3: Average response time based on channels number.

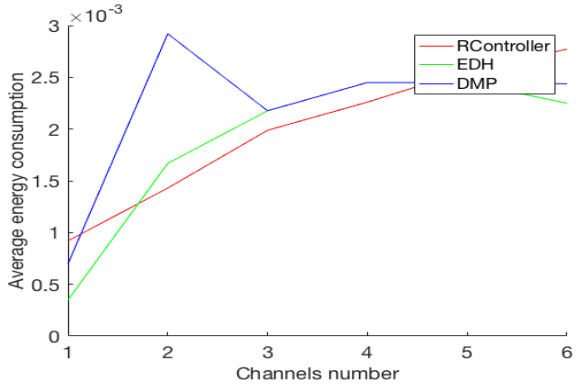


Figure 4: Average energy consumption based on channels number.

existing methods. It achieves the highest percentage of messages added with a lower response time and energy consumption.

CONCLUSION

In this paper, we develop a new scheduling mechanism for multi-channel RWSN (RController) to send messages reliably and timely with lower average in energy consumption and response time, where RController makes sure that the system is time-feasible and energy-feasible to send messages. Our conducted experiment shows that RController achieves significant improvements, since it has the highest percentage of successfully added messages that can go up to 85% while meeting deadlines with a lower average in response time and energy consumption. As a future work, we plan to extend the proposed approach to heterogeneous real-time RWSN with an advanced network topology and develop a distributed scheduling in those reconfigurations.

REFERENCES

Abdul-Salaam G.; Abdullah A.H.; Anisi M.H.; Gani A.; and Alelaiwi A., 2016. *A comparative analysis of energy conservation approaches in hybrid wireless sensor networks data collection protocols. Telecommunication Systems*, 61, no. 1, 159–179.

Chipara O.; Lu C.; and Roman G.C., 2013. *Real-Time Query Scheduling for Wireless Sensor Networks. IEEE Transactions on Computers*, 62, no. 9, 1850–1865.

Ferdous R.M.; Reza A.W.; and Siddiqui M.F., 2016. *Renewable energy harvesting for wireless sensors using passive RFID tag technology: a review. Renewable and Sustainable Energy Reviews*, 58, 1114–1128.

Ghribi I.; Abdallah R.B.; Khalgui M.; Li Z.; Alnowibet K.; and Platzner M., 2018. *R-Codesign: Codesign Methodology for Real-Time Reconfigurable Embedded Systems Under Energy Constraints. IEEE Access*, 6, 14078–14092.

Housseyni W.; Mosbahi O.; Khalgui M.; Li Z.; Yin L.; and Chetto M., 2018. *Multiagent Architecture for Distributed Adaptive Scheduling of Reconfigurable Real-Time Tasks With Energy Harvesting Constraints. IEEE Access*, 6, 2068–2084.

Kurt S.; Yildiz H.U.; Yigit M.; Tavli B.; and Gungor V.C., 2017. *Packet Size Optimization in Wireless Sensor Networks for Smart Grid Applications. IEEE Transactions Industrial Electronics*, 64, no. 3, 2392–2401.

Lee R.Y., 2011. *Computers,Networks, Systems, and Industrial Engineering 2011*, Springer, Berlin, Heidelberg, *Studies in Computational Intelligence*, vol. 365, chap. 18. 211.

Nasser N.; Karim L.; and Taleb T., 2013. *Dynamic Multilevel Priority Packet Scheduling Scheme for Wireless Sensor Network. IEEE Transactions on Wireless Communications*, 12, no. 4, 1448–1459.

Respondek J.S., 2010. *Numerical simulation in the partial differential equation controllability analysis with physically meaningful constraints. Mathematics and Computers in Simulation*, 81, no. 1, 120–132.

Stankovic J.A.; Spuri M.; Ramamritham K.; and Buttazzo G.C., 2012. *Deadline scheduling for real-time systems: EDF and related algorithms*, vol. 460. Springer Science & Business Media.

Wu S.; Niu J.; Member S.; Chou W.; and Guizani M., 2016. *Delay-Aware Energy Optimization for Flooding in Duty-Cycled Wireless Sensor Networks. IEEE Transactions on Wireless Communications*, 15, no. 12, 8449–8462.

Zhang J.; Khalgui M.; Li Z.; Mosbahi O.; and Al-Ahmari A.M., 2013. *R-TNCES: A Novel Formalism for Reconfigurable Discrete Event Control Systems. IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43, no. 4, 757–772.

MULTI-AGENT MODEL OF INFRASTRUCTURAL RETURN FOR AN INTERMEDIARY SERVICE PROVIDER

Anton Ivaschenko
Samara State Technical University
Molodogvardeyskaya street, 244
Samara, Russia
E-mail: anton.ivashenko@gmail.com

Stanislav Korchivoy
Radio Research and Development Institute
Kazakova str., 16
Moscow, Russia
E-mail: kor28@mail.ru

KEYWORDS

Multi-agent Technology, Digital Economics, Intermediary Service Provider, Infrastructural Return.

ABSTRACT

There is proposed an approach to increase the efficiency of IT infrastructure of Internet based intermediary services provider. The model of infrastructural return is based on formalization of products and facilities in the form of interrelated services that require single or multiple actions of certain costs. Implementation of these services based on the considered digital platform is presented in the form of a network that contains the objects of IT infrastructure, service providers and providing services interlinked by the relations of infrastructural supply and implementation. Using this model there is formalized an effect of services emission that describes the process of generation of new services. Analysis of the software platform efficiency is based on calculation of its implementation costs correlated with the income generated by the provided services. The proposed model was used in practice as a part of Internet based virtual intermediary operator, that provides digital services in transportation logistics.

INTRODUCTION

Modern trends of software development involve developing Internet based platforms open for suppliers and buyers of digital services. These platforms provide information for both parties and thus help them to find the best option of cooperation that meets all the requirements. For example, such services are popular in transportation logistics and taxi industry: a company with no own resources starts playing a mediator role by matching service providers and customers and generating benefits from data management and optimization. From this perspective there can be stated a new problem of software development for an intermediary service provider capable of permanent attracting the user's interest and generating sustainable profit under that changing conditions. In this paper there is proposed a formal model and solution for this problem based on multi-agent technology.

STATE OF THE ART

The concept and implementation of Intermediary service provider was presented in (Ivaschenko 2017). Most prominent results were achieved in transportation and service industry in the form of 5PL (Fifth Party Logistics) solution (Ivaschenko 2014) used by transportation companies to provide the best services for retail business. 5PL platform is open for transportation companies and even drivers and helps them negotiate with customers in integrated information space. 5PL provider owns no transportation resources itself but makes available a special service able to link suppliers and buyers. This service is based on the IT infrastructure, which plays the general role in 5PL business.

The challenges of development of Internet software solutions that act a mediator role are explored in (Hickson 2008, Machiraju 2002). In these papers intermediary layers of Internet services are being considered from technical prospective exploring various architectures of an overlay for federated service management, or web services management network. This concept relies on a network of communicating service intermediaries, each such intermediary being a proxy positioned between the service and the outside world.

Multi-agent technology (Gorodetskii 2012, Wooldridge 2002) is one of the successfully proven approaches that can be used for modeling and simulation of complex systems with high uncertainty, independent behavior of involved parties and self-organization. Since the users of intermediary service provider are autonomous entities driven by individual economic interests, multi-agent models can become effective to simulate possible negotiations and forecast beneficial market strategies.

Modern online web services provide functionality capable to track all the events that characterize the processes of Internet users' interaction and thus determine and study the basic trends of social and economic systems evolution. Recent studies in this area (Kadushin 2012, One Internet 2016, Wei 2016) propose several relevant models that can be used to describe the principles of Internet users' activity and utilize them in various problem domains including e.g. targeted online advertising and social research.

THE MODEL OF INFRASTRUCTURAL RETURN

Based on analysis of organizational and technological aspects of Intermediary service provider implementation there is proposed an original model of infrastructural return. Let us designate by s_i a certain digital service that

consumers u_k can use. The facts of consumption of the service will be denoted in the form of events:

$$e_{i,j,k} = e_{i,j,k}(s_i, u_k, c_{i,j,k}, \{c'_{i,j,k,l}\}, t_{i,j,k}), \quad (1)$$

where, $t_{i,j,k}$ is the time of occurrence of the event, j is the index of circulation for consumption, $c_{i,j,k}$ represents the cost of the service for the end user at the time $t_{i,j,k}$, $\{c'_{i,j,k,l}\}$ is a general estimate of the service costs, including the costs of building of its infrastructure, depreciation, insurance, production, administrative and commercial expenses, interest on loans, income taxes, etc..

Each service for its implementation requires one or more providers d_n and appropriate software IT infrastructure. Within the framework of the IT infrastructure used, there are utilized corresponding objects h_m with specific costs v_m and availability period $\Delta\tau^*_m$.

The events of putting IT infrastructure facilities into operation will be denoted by a Boolean function:

$$q_m = q_m(h_m, v_m, \{v'_{m,l}\}, \Delta\tau^*_m) = \{0,1\}, \quad (2)$$

where $\{v'_{m,l}\}$ presents the costs of an object h_m development and maintenance.

Therefore, implementation of digital services within an Intermediary service provider can be presented in the form of a graph linking IT infrastructure objects, service providers and produced services on the top, with each other using corresponding relationships of services infrastructural support $b_{i,m}$ (s_i, h_m), emission (the receipt of a new service based on the existing ones), and execution.

Using the introduced definitions, it is possible to determine the cost of the IT infrastructure of the Intermediary service provider and the costs of its operating over a period $\Delta\tau_n = (t'_n, t'_n + \Delta t'_n)$, $n = 1..N$:

$$V(\Delta\tau_n) = \sum_m \sum_{i,j,k} q_m \cdot v_m \cdot \delta(t_{i,j,k} \in \Delta\tau_n), \quad (3)$$

where $\delta(x) = \begin{cases} 1, x = true, \\ 0, otherwise. \end{cases}$

Revenues from the sale of services (net of charges) are:

$$C(\Delta\tau_n) = \sum_{i,j,k} \left(e_{i,j,k} \cdot \delta(t_{i,j,k} \in \Delta\tau_n) \cdot \left(c_{i,j,k} - \sum_l c'_{i,j,k,l} \right) \right). \quad (4)$$

These indicators can be used to estimate net present value:

$$NPV = -V(\Delta\tau_0) + \sum_{n=1}^{N} \frac{(C(\Delta\tau_n) - V(\Delta\tau_n))}{(1+I)^n}, \quad (5)$$

where I is the discount rate, $\Delta\tau_n$ is the annual period, $V(\Delta\tau_0)$ is the investment expenditure for the initial implementation of IT infrastructure.

The main difference between a digital platform and conventional IT solutions for Intermediary service operator lies in the possibility of providing new services not covered for by the original specification. This happens by the appearance of new types of services by combining existing ones using available suppliers and infrastructure.
This feature can be described by the effect of services emission, when a subset of existing services and their purchase events lead to the emergence of new services using the same IT infrastructure. Therefore, one should strive to generate services emission and maximize secondary services when building an IT infrastructure of a digital platform. This will help to increase NPV by a value of infrastructural return:

$$R(\Delta\tau_n) = \sum_{i,j,k} \left(e_{i,j,k} \cdot \delta(t_{i,j,k} \in \Delta\tau_n) \cdot \right. \\ \left. \cdot \delta\left(\sum_m b_{i,m}(s_i, h_m) = 0 \right) \cdot \left(c_{i,j,k} - \sum_l c'_{i,j,k,l} \right) \right) \quad (6)$$

With the accumulation of enough services, they can be combined and supplemented, generating new services that do not require additional resources. New services appear under the influence of their users' activity. For example, in the case of frequent and regular sharing of two or more services by different users, they can be combined as a complex service. Thus, the issue of services is supported, supported by consumption and arising in addition to the costs incurred.

MULTI-AGENT SOLUTION

Modeling and simulation of services emission powered by the described above effect of infrastructural return can improve the quality of intermediary service provider. This solution can be described and implemented using a multi-agent architecture described by Fig. 1. This solution is made by analogy to multi-agent platforms developed to study social media (Ivaschenko 2018). The main idea is to describe a set of concepts for each actor (service provider or customer) including <focus, context, and overlay context> and use it for approximation of its behavior. Focus is used to represent the current interest of Internet platform users. Context is used to formalize informational space in which the agent performs its negotiating activity.

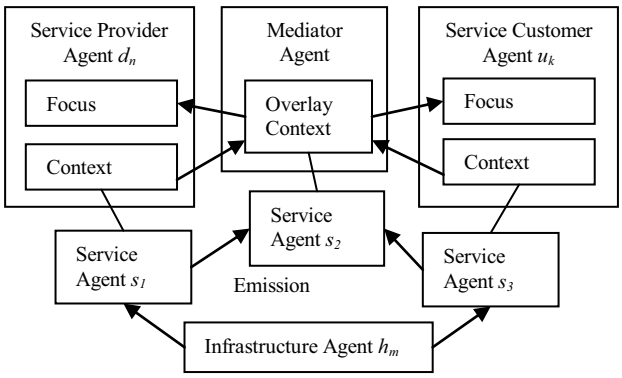


Figure 1: Multi-agent Solution

To modify the focus and attract the actor's interest to specific services there can be automatically generated an additional "overlay context" in the form of hints or notifications. The context of the agents contains services that have been successfully used in past representing the agents' experience.

Using this data, a Mediator agent can generate new services using the same IT infrastructure. These services should be included into the focus of actor agents to stimulate consumption and thus support the emission. Changes of actor's focus represent the evolution of their interest and can be used to simulate and analyze the directions of possible Intermediary service provider improvements. Behavior of each actor can be determined by a combination of several concurrent interests, nevertheless all of them will be reflected in corresponding focus changes.

Context changes are correlated with the actor's focus modification. The focus cannot be considerably new to provide positive perception, and at the same time it is not equal to the context to be able to excite interest. Identification of the context and focus for the agents of services providers and customers and additional overlay context on their basis is proposed to use to manage their interest by generating and advertising of new services using existing resources of IT infrastructure.

IMPLEMENTATION RESULTS

Implementation of the Intermediary service provider based on the proposed multi-agent model is presented in Fig. 2. In addition to the service provider itself and corresponding IT infrastructure (including computers, applications and online Web services) it is proposed to introduce a control unit capable of monitoring, simulation and management. This unit can capture and track the data of platform popularity and use, analyze the efficiency of IT infrastructure utilization and promptly recommend new features that need to be implemented to remain in-demand. Such an approach allows improving the quality of Internet based systems software development and testing for virtual intermediary operators in service industry at the stages of investment and evaluation of the projects in digital economics. Specific features of such solutions are concerned with virtualization of services in digital social and economic systems.

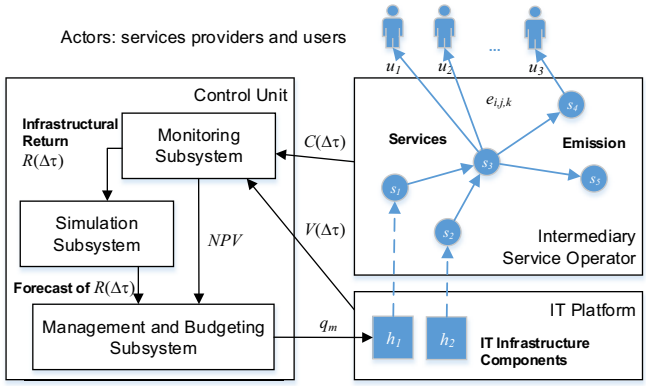


Figure 2: Implementation Architecture

Fig. 3. presents the difference between regularly provided services consumption with (lower curve) and without (upper curve) services emission. Here is demonstrated the dependency of cost (money equivalent) to time. There are generated up to 100 services daily consumption events with the possible emission of 10 new proposals.
New services do not require any additional software infrastructure.

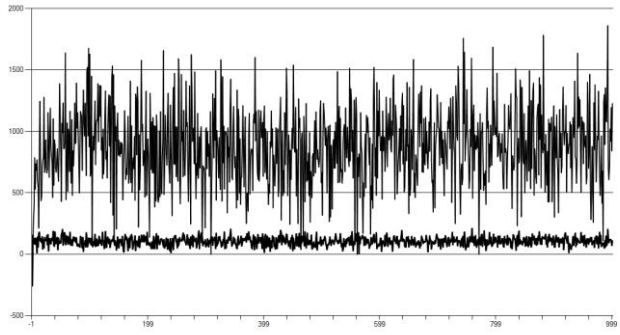


Figure 3: Simulation Results

CONCLUSION

In this paper there is proposed a multi-agent model of infrastructural return for an Intermediary service provider. The proposed model allows improving the coordination of virtual benefits of digital economics and its software development expenses. Analysis of the software platform efficiency is based on calculation of its implementation costs correlated with the income generated by the provided services. The proposed model can be recommended to be used in practice as a part of Internet based virtual intermediary operator, that provides digital services.

REFERENCES

Gorodetskii, V.I.. 2012. "Self-organization and multiagent systems: I. Models of multiagent self-organization", *Journal of Computer and Systems Sciences International*, vol. 51, issue 2. 256-281

Hickson, A.; B. Wirth, and G. Morales. 2008. “Supply chain intermediaries study”, University of Manitoba Transport Institute, 56 p.

Ivaschenko, A. 2014. “Multi-agent solution for business processes management of 5PL transportation provider”. *Lecture Notes in Business Information Processing*, Vol. 170. 110-120.

Ivaschenko, A.; I. Syusin, amd P. Sitnikov. 2017. “Cooperative decision making modeling in transportation logistics dispatching system”, *Proceedings of the 2017 European Simulation and Modeling Conference (ESM 2017)*, Lisbon, Portugal, EUROSIS-ETI. 227-232

Ivaschenko A.; A. Khorina, V. Isayko, D. Krupin, P. Sitnikov. 2018. “Online creativity modeling and analysis based on Big Data of social networks”. Proceedings of Computing Conference 2018, London, UK. 199-202

Kadushin, C. 2012. “Understanding social networks: theories, concepts, and findings”, Oxford University Press, 264

Machiraju, V.; A. Sahai, and A. Moorsel. 2002. “Web services management network: an overlay network for federated service management”. Hewlett-Packard Laboratories, Hpl hp techreports, 234 p.

One Internet. 2016. Global commission on Internet Governance. Report. 138 p.

Wei, W.; K. Joseph, H. Liu, K.M. Carley. 2016. “Exploring characteristics of suspended users and network stability on Twitter”. *Social Network Analysis and Mining*, 6:51.

Wooldridge, M. 2002. “An introduction to multi-agent systems”, John Wiley and Sons, Chichester. 340 p.

ENVIRONMENTAL SIMULATION

STOCHASTIC MODEL OF THE JOINT TIME-SERIES OF AIR TEMPERATURE AND ATMOSPHERIC PRESSURE

Nina A. Kargapolova
Institute of Computational Mathematics and Mathematical Geophysics SB RAS
Pr. Lavrent'eva 6
630090 Novosibirsk
Russia

Novosibirsk State University
Pirogov St. 1
630090 Novosibirsk
Russia

E-mail: nkargapolova@gmail.com

KEYWORDS
Stochastic Simulation, Time-series Analysis, Non-stationary Random Process, Periodically Correlated Process, Air Temperature, Atmospheric Pressure, Model Validation.

ABSTRACT

In this paper a numerical stochastic model of the joint non-stationary time-series of the air temperature and atmospheric pressure is proposed. The model is based on an assumption that real weather processes are periodically correlated random processes with a period equal to 1 day. This assumption takes into account the diurnal variation of real meteorological processes, defined by the day/night alternation. The input parameters of the model (one-dimensional distributions of the air temperature and atmospheric pressure and the correlation structure of the joint time-series) are determined from the data of long-term real observations at weather stations.

INRTODUCTION

The study of properties of atmospheric processes involving adverse weather conditions (for example, long-term heavy precipitation, dry hot wind, unfavorable combination of low temperature and high relative humidity, long periods of high temperature, precipitation absence and strong wind that foster a spread of wildland fires, etc.) is of great scientific and practical importance. Results of this study are crucial for solution of some problems in agroclimatology, structural engineering, planning of heating and conditioning systems and in many other applied areas (see, for example, Araya and Kisekka 2017, Ballesteros-Pérez et al. 2015, Cardil et al. 2013, Pall et al. 2013).

In the design of heating, ventilation and air conditioning systems, climatic characteristics of air temperature and relative humidity are used, which are contained in regulation documents. It is known, however that the available information gives a very rough idea of the peculiarities of the temperature-humidity combination in a particular locality and does not correspond to the current trend of energy saving. The problem becomes particularly relevant in connection with the ongoing climate changes, which show growing demands for energy consumption in the summer season. For the optimal functioning of air conditioning systems, it is desirable to have an idea of the detailed characteristics of changes and variability in the atmospheric air indices, for example, such an index as the enthalpy of moist air (Khomutskiy 2017). The enthalpy of moist air is a non-linear function of atmospheric pressure, air temperature and relative humidity. It is possible to study some statistical properties of the enthalpy on the basis of real observation data collected at weather stations. But, unfortunately, there are extremely few real data for obtaining stable statistical characteristics of rare / extreme behavior of the enthalpy. Moreover, the behavior of these characteristics is influenced by climatic changes, and hence it is not always possible to obtain reliable estimates only from observation data. In this regard, it is necessary to construct a stochastic model of the joint time-series of atmospheric pressure, air temperature and relative humidity that reproduces accurately the main properties of the real weather processes, and then to study properties of the enthalpy on a basis of simulated trajectories. If the model is a parametric one, it is possibly to vary parameters of the model (for example, average temperature, correlations between pressure and humidity, etc.) and study dependence of the enthalpy on these parameters. This is a typical approach to study of rare / extreme weather events and interdependence of weather processes (see, for example, Ailliot 2015, Evstafieva et al. 2005, Kargapolova 2017, Kleiber 2013, Ogorodnikov 2013, Semenov 2002).

The first step of such model construction is presented in (Kargapolova et al. 2018). In that paper an algorithm for simulation of the joint time-series of air temperature and relative humidity is presented. In this paper the second step of construction of a stochastic model of the enthalpy time-series is presented. A numerical stochastic model of the joint non-stationary time-series of the air temperature and atmospheric pressure is proposed. The model is based on an assumption that real weather processes are periodically correlated random processes with a period equal to 1 day. This assumption takes into account the diurnal variation of real meteorological processes, defined by the day/night alternation. The input parameters of the model (one-dimensional distributions of the air temperature and atmospheric pressure and the correlation structure of the joint time-series) are determined from the data of long-term real observations at weather stations.

REAL DATA

To study statistical properties of real weather processes and to define parameters of a stochastic model real data collected at weather stations 8 times per day (i.e. every 3 hours) during

23 years from 1993 to 2015 were used. The model proposed in this paper was validated on a basis of real data from weather stations located in different climatic zones (for example, subtropical zone, temperate continental zone, polar zone, etc.). Although all examples in the article are given only for stations in the cities of Sochi (subtropical zone) and Tomsk (continental-cyclonic zone), all conclusions are valid for all considered weather stations.

The most noticeable feature of the time-series of the air temperature and atmospheric pressure at short time-intervals (week-, decadelong intervals) is the diurnal variation, defined by the day/night alternation. It is shown in (Kargapolova 2018) that air temperature time-series on such time intervals could be considered as periodically correlated random processes with a period equal to 24 hours. The same is valid for the time-series of atmospheric pressure. Recall that a random process $X(t)$ is a periodically correlated process with a period T if its mathematical expectation, variance and correlation function are periodic functions (Dragan et al. 1987):

$$EX(t) = EX(t+T),\ DX(t) = DX(t+T),$$
$$corr\left(X(t_1), X(t_2)\right) = corr\left(X(t_1+T), X(t_2+T)\right).$$

Fig. 1 and Fig. 2 represent mean values and variance of air temperature and atmospheric pressure estimated on the real data under assumption that time-series are periodically correlated on a 10-days time-interval.

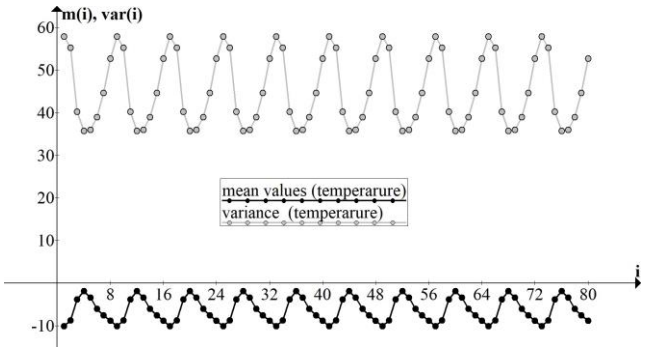


Figure 1: Mean Values (°C) and Variance of Air Temperature as Functions of Time. Tomsk, March

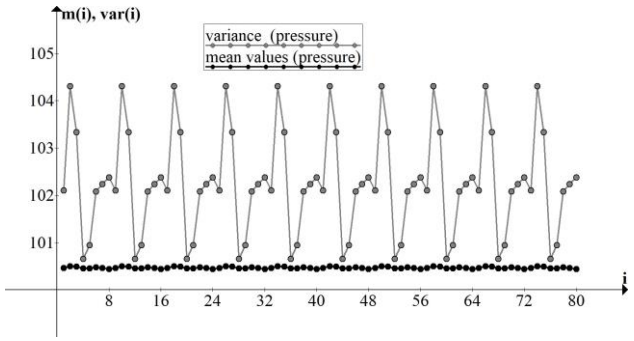


Figure 2: Mean Values (MPa) and Variance of Atmospheric Pressure as Functions of Time. Tomsk, March

It should be noted that on long time intervals (like month-, season-, yearlong intervals) mean values and variance of both weather elements still oscillate, but influence of a seasonal variation doesn't let to consider these statistical functions as periodic ones.

MODEL INPUT PARAMETERS

Let consider joint meteorological time-series $\vec{T} = (T_1, T_2, \ldots, T_{8d})$ of air temperature and $\vec{H} = (H_1, H_2, \ldots, H_{8d})$ of atmospheric pressure as periodically correlated discrete-time random processes with a period $T = 8$, where T_i and P_i are air temperature and atmospheric pressure at a measurement number i ("at a time moment i") respectively, d is a number of days (it is a length of the selected time-interval).

One-dimensional distributions and a correlation structure of the joint time-series of air temperature and atmospheric pressure are used as the model input parameters.

To construct a stochastic model, the use of sample one-dimensional distributions is not advisable, since the sample distributions do not have any tails, and therefore do not allow one to estimate the probability of occurrence of extreme values of meteorological elements. In this connection it is necessary to approximate the sample distributions densities by some analytic densities, which, on the one hand, do not greatly alter the form of a sample distribution and its moments, and on the other – possess tails.

Numerical experiments show that mixtures

$$g_k(x) = \theta_k \frac{1}{b_{k1}\sqrt{2\pi}} \exp\left(-\frac{(x-a_{k1})^2}{2b_{k1}^2}\right) +$$
$$+(1-\theta_k)\frac{1}{b_{k2}\sqrt{2\pi}} \exp\left(-\frac{(x-a_{k2})^2}{2b_{k2}^2}\right),$$
$$0 \le \theta_k \le 1,\ k = \overline{1,8}.$$

of 2 Gaussian distributions closely approximate sample histograms $s_k(x)$ of air temperature for all measurements $k = \overline{1,8}$ (and, therefore, for all moments $i = 1,2,\ldots,8d$) at all the considered weather stations. Parameters $\theta_k, a_{k1}, b_{k1}^2, a_{k2}, b_{k2}^2$ were chosen using an algorithm, proposed in (Marchenko and Minakova 1980). This algorithm let to choose such parameters of a mixture $g_k(x)$ (and corresponding CDF $G_k(x)$) that mathematical expectation, variance and skewness of a random variable with a density $g_k(x)$ are equal to corresponding sample characteristics and function $g_k(x)$ minimizes the Pearson's functional, that describes difference between $s_k(x)$ and $g_k(x)$. For each k sample mathematical expectation, variance and skewness were estimated on a basis of $23d$-element sample. The numerical experiments show that the coefficients of excess kurtosis of the approximating densities are close to sample ones.

It's much more difficult to approximate histograms of atmospheric pressure. The problem is that type of distribution of this weather element varies greatly from one weather station to another. At some weather stations, the distribution is unimodal, on other ones – it is bimodal. The skewness and excess kurtosis vary also over a wide range. For most of the considered weather stations practically same quality of approximation is achieved by approximating of the sample distributions with the mixtures of 2 Gamma-distributions and with the mixtures of 2 truncated to $[0,+\infty)$ Gaussian distributions. For these stations to simplify the simulation mixtures of 2 truncated Gaussian distributions were used. Parameters of these mixtures were chosen on a basis of the EM algorithm (see, for example, Dempster et al. 1977). For the weather stations located in the subarctic zone, both mentioned families of distributions give unacceptable quality of approximation. For these stations relatively good quality of approximation is achieved by the F-distribution. Let $b_k(x),\ k=\overline{1,8}$ denote a pdf of atmospheric pressure (either the mixture of 2 truncated to $[0,+\infty)$ Gaussian distributions or the F-distributions) and $B_k(x),\ k=\overline{1,8}$ are corresponding CDFs.

For simulation of the joint time-series sample correlation matrices were used. Analysis of real data shows that for all meteorological stations and time-intervals considered, the amplitudes of diurnal oscillations of the auto- and cross-correlation functions of air temperature and atmospheric pressure are significant. As an illustration, Fig. 3 represents sample cross-correlation coefficients $corr(\mathrm{T}_i,\mathrm{P}_{i+h})$.

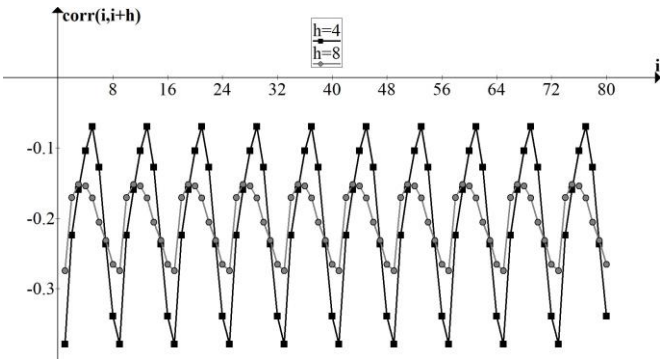


Figure 3: Sample cross-correlation coefficients $corr(\mathrm{T}_i,\mathrm{P}_{i+h})$.Tomsk, June

To construct a parametric stochastic model of the joint time-series it is necessary to approximate the sample correlation function of the periodically correlated process with some analytic parametric function (like it was done with one-dimensional distributions). Such approximation is a work in progress.

SIMULATION ALGORITHM

It is necessary to simulate a sequence $(\vec{T},\vec{P})=(\mathrm{T}_1,\mathrm{T}_2,\ldots,\mathrm{T}_{8d},\mathrm{P}_1,\mathrm{P}_2,\ldots,\mathrm{P}_{8d})$, which has the following properties:

1) the one-dimensional distribution densities of the components T_i and P_i are $g_k(x)$ and $b_k(x)$ respectively, where $i=8j+k,\ j=\overline{1,d},\ k=\overline{1,8}$;

2) the sequence $(\vec{T},\vec{P})$ has the correlation matrix

$$R=\begin{pmatrix} R_T & R_{TP} \\ R_{PT} & R_P \end{pmatrix},$$

where R_T, R_P are sample autocorrelation matrices of air temperature and atmospheric pressure respectively, and R_{TP}, R_{PT} are sample cross-correlation matrices of these two weather elements. It should be noted that, although the matrices R_T, R_P, R_{TP}, R_{PT} are block-Toeplitz matrices, the matrix R is not block-Toeplitz. For simulation of $(\vec{T},\vec{P})$ with given one-dimensional distributions and given correlation matrix a method of inverse distribution function was used (Piranashvili 1966; Ogorodnikov and Prigarin 1996). In the framework of this method, simulation of the sequence $(\vec{T},\vec{P})$ comes down to an algorithm with 3 steps:

1. Calculation of a matrix R' that is a correlation matrix of an auxiliary standard Gaussian process $(\vec{T}',\vec{P}')$. Element $r'(\mathrm{i},\mathrm{j})\ i,j=\overline{1,2\times 8d}$ of the matrix R' is a solution of the equation

$$r(\mathrm{i},\mathrm{j})=$$
$$=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F_i^{-1}\left(\Phi(x)\right)F_j^{-1}\left(\Phi(y)\right)\varphi\left(x,y,r'(\mathrm{i},\mathrm{j})\right)dxdy,$$

where $r(\mathrm{i},\mathrm{j})$ is an element of the matrix R corresponding to $r'(\mathrm{i},\mathrm{j})$, the function

$$\varphi\left(x,y,r'(\mathrm{i},\mathrm{j})\right)=$$
$$=\left[2\pi\sqrt{1-\left(r'(\mathrm{i},\mathrm{j})\right)^2}\exp\left(\frac{2r'(\mathrm{i},\mathrm{j})xy-x^2-y^2}{2\left(1-\left(r'(\mathrm{i},\mathrm{j})\right)^2\right)}\right)\right]$$

is a distribution density of a bivariate Gaussian vector with zero mean, variance equal to 1 and correlation coefficient $r'(\mathrm{i},\mathrm{j})$ between components number i and j, $\Phi(\cdot)$ is a CDF of a standard normal distribution, G_i, B_j are CDFs corresponding to distribution densities of air temperature and atmospheric pressure.

2. Simulation of the standard Gaussian sequence $(\vec{T}',\vec{P}')$ with the correlation matrix R'.

3. Transformation of $(\vec{T}',\vec{P}')$ into $(\vec{T},\vec{P})$:

$$\mathrm{T}_\mathrm{i}=G_\mathrm{i}^{-1}\left(\Phi\left(\mathrm{T}_\mathrm{i}'\right)\right),\ \mathrm{i}=\overline{1,8\mathrm{d}},\ \mathrm{P}_\mathrm{j}=B_\mathrm{j}^{-1}\left(\Phi\left(\mathrm{P}_\mathrm{j}'\right)\right),\ \mathrm{j}=\overline{1,8\mathrm{d}}.$$

Steps 3 and 4 are repeated as many times as many trajectories are required. It should be noted that, due to the block-Toeplitz structure of the matrices R_T, R_P, R_{TP}, R_{PT}, on the first step of the simulation algorithm to define fully the $(2\times 8d)\times(2\times 8d)$ matrix R', it is enough to solve only $3\times 8\times 8d$ equations for $r'(\mathrm{i},\mathrm{j})$.

If the matrix R', obtained in the first step, is not positively defined, it must be regularized. Several methods of regularization are described in (Ogorodnikov and Prigarin, 1996). In this paper a method of regularization based on substitution of negative eigenvalues of the matrix R' with small positive numbers was used. In most cases after regularization the matrix R' requires normalization.

Simulation of the standard Gaussian sequence $(\vec{T}', \vec{P}')$ with correlation matrix R' in the second step could be done using Cholesky or spectral decomposition of the matrix R'. The last one was used in this paper.

NUMERICAL EXPERIMENTS

Because of the specific nature of the inverse distribution function method, the correlation structure of the simulated process differs from the correlation structure of the real one. This is explained mainly by 2 reasons. First reason is that regularization and normalization of the matrix R' alter the correlation structure of the process $(\vec{T}', \vec{P}')$ and, therefore, the correlation structure of the simulated joint time-series. Moreover, transformation of $(\vec{T}', \vec{P}')$ into $(\vec{T}, \vec{P})$ brings additional distortion in the correlation matrix because of computational errors in computation of the CDF of a standard normal distribution and numerical inversion of the distribution functions G_i, B_j, $i, j = \overline{1,8}$. Nevertheless, it should be noted that this difference is rather small, which, for example, is illustrated in Fig. 4. Here and below 10^5 simulated trajectories were used for estimations. To denote estimations based on real and simulated data, abbreviations RD and PCM are used respectively.

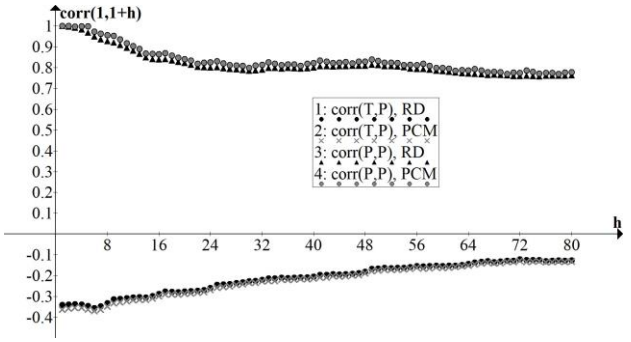


Figure 4: Correlation coefficients $corr(\mathrm{T}_1, \mathrm{P}_{1+h})$ estimated on real and simulated data (curves 1 and 2 respectively) and correlation coefficients $corr(\mathrm{P}_1, \mathrm{P}_{1+h})$ estimated on real and simulated data (curves 3 and 4 respectively). Sochi, August

Any stochastic model has to be verified before one starts to use simulated trajectories to study properties of a simulated process. For a model verification, it is necessary to compare simulated and real data based estimations of such characteristics, which, on the one hand, are reliably estimated by real data, and on the other hand are not input parameters of the model. Here are several examples of such characteristics.

First characteristic that was considered was "probability P of a rapid change of atmospheric pressure". As a rapid change of pressure, a change for more than $\Delta\ hPa$ in less than n hours was considered (usually this characteristic is considered for $n \le 24$). Tab. 1 shows corresponding estimations for $n = 24$. For estimations based on the real data the 99% confidence intervals are given.

Another characteristic that was considered was "average number of days in a month with a minimum / maximum daily temperature below / above given level $l\ ^oC$". Tab. 2 shows an example of the corresponding estimations, obtained on a basis of real and simulated data. Differences between estimations on the real and simulated data are less than statistical errors of estimations on small samples of real data. It should be noted that here $d = 31$. The assumption about periodically correlated structure of the real processes on a monthlong interval is valid only for in-season months.

Table 1: Probabilities of the Atmospheric Pressure Rapid Change. Sochi. December, 1-15

Δ	RD $P \pm 3\sigma$	PCM
1	0.965±0.001	0.966
3	0.628±0.034	0.661
5	0.361±0.065	0.415
7	0.190±0.059	0.188
11	0.044±0.029	0.031
15	0.012±0.014	0.008
19	0.001±0.007	0.002

Table 2: Average Number of Days in a Month with a Maximum Daily Temperature Below Given Level $l\ ^oC$. Sochi

l	October		December	
	RD	PCM	RD	PCM
22	22.48	22.55	30.87	30.86
18	11.00	10.83	29.61	29.71
14	2.17	2.18	23.91	24.08
10	0.22	0.25	11.65	11.85
6	0.00	0.00	3.35	3.08
2	0.00	0.00	0.39	0.34
0	0.00	0.00	0.09	0.06

Real and simulated data based estimations of probabilities of the event "air temperature is below / above the given level $lt\ ^oC$ and atmospheric pressure is below / above the given level $lp\ hPa$" were also compared. For verification, levels

lt, lp close to the mean values of air temperature and atmospheric pressure were chosen (see Tab. 3).
Tab. 4 shows simulated and real data based estimations of the average daily temperature $A_i\ ^oC$ in a day number i. For most of the considered time-intervals, simulated data based estimations lay in the corresponding intervals $(A_i - \sigma, A_i + \sigma)$.

Table 3: Probabilities of the Event "Air Temperature is Below the Given Level $lt\ ^oC$ and Atmospheric Pressure is Above the Given Level lp ". Sochi. June, 1-15

lt	$lp = 992$		$lp = 1002$		$lp = 1012$	
	RD	PCM	RD	PCM	RD	PCM
24	0.804	0.802	0.478	0.480	0.102	0.109
20	0.417	0.426	0.268	0.282	0.069	0.073
16	0.091	0.090	0.073	0.071	0.039	0.025
12	0.005	0.007	0.005	0.007	0.003	0.003

Table 4: Average Daily Temperature $A_i\ ^oC$ in a Day Number i. Sochi. February, 1-10

i	RD $A_i \pm \sigma$	PCM
1	5.205±0.826	5.908
3	5.373±0.850	5.935
5	5.276±0.880	5.907
7	6.836±0.871	5.926
9	6.943±0.858	5.911

CONCLUSION

In general, the model proposed reproduces quite well the properties of real joint time-series of air temperature and atmospheric pressure on short time-intervals. Numerical experiments show that simulated trajectories could be used for study of the properties of extreme and anomalous meteorological events (for example, for estimating the probabilities of sudden temperature / pressure changes or long, unfavorable periods of high temperature and low pressure). For simulation of time-series on longer time-intervals (like month-, season-, yearlong intervals) it is better to use a nonstationary model which let to take into account not only daily, but also seasonal variation of the weather elements. As it is shown in (Kargapolova 2018) for the air temperature time-series, simulation of the non-stationary processes is more time-consuming procedure in comparison with simulation of the periodically correlated processes. First, it is necessary to approximate $2\times 8d$ distribution densities of the joint time-series of air temperature and atmospheric pressure (instead of 2×8 densities in case of the considered in this paper periodically correlated model). Second, correlation matrix of a non-stationary process doesn't have any specific features. It means that in the framework of inverse distribution function method it is necessary to solve $8d\times(2\times 8d-1)$ equations for $r'(\mathrm{i},\mathrm{j})$.
Numerical implementation of the non-stationary model of the joint time-series of air temperature and atmospheric pressure is the next step in a study of properties of weather processes under consideration.
In the future, the model presented will be expanded – instead of the joint time-series of air temperature and atmospheric pressure, the joint series of air temperature, atmospheric pressure and air relative humidity will be simulated. Simulation of such a weather combination is of interest, because based on the simulated trajectories it is possible to study the properties of the time-series of different air heat content indicators.

ACKNOWLEDGEMENTS

Author is deeply indebted to Prof. V. Ogorodnikov for his help and fruitful discussions.
This work was partly financially supported by the Russian Foundation for Basis Research (grant No 18-01-00149-a) and the President of the Russian Federation (grant No MK-659.2017.1).

REFERENCES

Ailliot P., D. Allard, V. Monbet and P. Naveau. 2015. "Stochastic weather generators: an overview of weather type models". In *Journal de la Société Française de Statistique*, No 1 (156), 101-113.

Araya A. and L. Kisekka. 2017. "Evaluating the impact of future climate change on irrigated maize production in Kansas". In *Climate Risk Management*, Vol. 17, 139-154.

Ballesteros-Pérez P., M. L. del Campo-Hitschfeld, M.A. González-Naranjo, M.C. González-Cruz. 2015. "Climate and construction delays: case study in Chile". In *Engineering, Construction and Architectural Management*, Vol. 22, No 6, 596-621.

Cardil A., D. M. Molina, J. Ramirez and C. Vega-García. 2013. "Trends in adverse weather patterns and large wildland fires in Aragon (NE Spain) from 1978 to 2010". In *Nat. Hazards Earth Syst. Sci.* No 13, 1393–1399.

Dempster A. P., N. M. Laird and D. B. Rubin. 1977. "Maximum likelihood from incomplete data via the the EM algorithm (with discussion)". In *J. of the Royal Stat. Society B*, Vol. 39, 1-38.

Dragan Ya. P., V. A. Rozhkov and I. N. Yavorskii. 1987. *Methods of probabilistic analysis of rhythmic of oceanology processes*. Gidrometeoizdat, Leningrad. (in Russian)

Evstafieva A. I., E. I. Khlebnikova and V. A. Ogorodnikov. 2005. **"**Numerical stochastic models for complexes of time series of weather elements". In *Russ. J. Num. Anal. Math. Modelling,* Vol. 20, No 6, 535–548.

Kargapolova N.A. 2017. "Monte Carlo Simulation of Daily Precipitation and River Flow Conditional Spatio-Temporal Fields". In *Proceedings of the 31st European Simulation and Modelling Conference 2017* (Lisbon, October 25-27), 311-315.

Kargapolova N. 2018. "Monte Carlo Simulation of Non-stationary Air Temperature Time-series". In *Proceedings of the SIMULTECH-2018* (Porto, July 29-31), 323-329.

Kargapolova N., E. Khlebnikova and V. Ogorodnikov. 2018. "Monte Carlo simulation of the joint non-Gaussian periodically correlated time-series of air temperature and relative humidity". In *Statistical Papers*, DOI 10.1007/s00362-018-1031-z.

Khomutskiy Ju. 2017. "Usage of observed temperature data for designing of air conditioning and ventilation systems". In *World of Climate*, Vol. 101, 156-161. (in Russian)

Kleiber W., R. W. Katz and B. Rajagopalan. 2013. "Daily minimum and maximum temperature simulation over complex terrain". In *Annals of Applied Statistics*, No 7, 588-612.

Marchenko A. S. and L. A. Minakova. 1980. “Probabilistic model of air temperature time-series”. In *Meteorology and Hydrology*, No. 9, 39-47. (in Russian)

Ogorodnikov V. A. 2013. *Stochastic models of meteorological processes: Teaching aid*. NSU, Novosibirsk.

Ogorodnikov V. A. and S. M. Prigarin. 1996. *Numerical Modelling of Random Processes and Fields: Algorithms and Applications.* VSP, Utrecht.

Pall R. K., V. K. Sehgal, A. K. Misra, K. Ghosh, U. C. Mohanty and R. S. Rana. 2013. “Application of Seasonal Temperature and Rainfall Forecast for Wheat Yield Prediction for Palampur, Himachal Pradesh”. In *Int. J. of Agriculture and Food Sc. Tech.*, Vol. 4, No 5, 453-460.

Piranashvili Z. A. 1966. “Some problems of statistical probabilistic modelling of random processes”. In *Probl. of Operations Res.*, Tbilisi, 53-91. (in Russian)

Semenov M. A. and E. M. Barrow. 2002. *LARS-WG: A Stochastic Weather Generator for Use in Climate Impact Studies. User Manual*. Harpenden.

AUTHOR BIOGRAPHY

NINA KARGAPOLOVA was born in Novosibirsk, Russia and went to the Novosibirsk State University (NSU), where she studied Mathematics and obtained her Bachelor and Master degrees in 2008 and 2010 respectively. After 3 years of postgraduate training at the Institute of Computational Mathematics and Mathematical Geophysics Siberian Branch of the Russian Academy of Science (ICM&MG SB RAS) she got a Doctor degree in Mathematics (area of competence – Computational Mathematics). Since 2013 she has been working as a researcher at the Laboratory of Stochastic Problems in ICM&MG SB RAS and at the same time she is employed in the Associate Professor position at NSU.

TEMPOROSPATIAL EPIDEMIC SIMULATIONS USING HETEROGENEOUS COMPUTING

Dhananjai M. Rao
CSE Department, Miami University
Oxford, OH 45056, USA.
email: raodm@miamiOH.edu

KEYWORDS

Temporospatial epidemic models, discrete event simulation, GPGPU, OpenCL, Heterogeneous Computing (HC)

ABSTRACT

Epidemiological analysis is conducted via simulation using complex temporospatial models. In such simulations, 50%–90% of simulation runtime is spent in solving equations used to model epidemic progression. Heterogenerous Computing (HC) using General Purpose Graphics Processing Units (GPGPUs) and CPUs hold considerable potential to reduce simulation times. However, significant differences in programming models of GPGPUs and CPUs hinder their effective use. Consequently, we have developed an epidemic modeling and heterogeneous simulation environment called MUSE-HC. In MUSE-HC, discrete event processing is performed on the CPU while epidemic equation processing is performed on a GPGPU. MUSE-HC also provides a domain-specific modeling language called Epidemic Description Language (EDL) along with compiler and libraries to streamline modeling for non-computing experts. Our experiments conducted using synthetic benchmarks show that our heterogeneous approach can improve simulation performance by up to 16× for certain classes of epidemic models.

INTRODUCTION & BACKGROUND

Multinational epidemics of communicable and zoonotic diseases such as Zika fever, Chikungunya, influenza, etc. continue to pose serious health and socioeconomic challenges. Hence, there is a heightened urgency to develop comprehensive methods for proactively containing epidemics. Epidemic containment strategies heavily rely on epidemiological modeling and simulation for disease forecasting and analysis. Since, multinational epidemics have complex dynamics that vary with geography and time, temporospatial models (see Figure 1) are used to characterize epidemic progressions. Such models are typically represented as a set of interacting "agents", where each agent models epidemic progression in a collocated population. Figure 1 shows an example of a temporospatial epidemic model with circular agents that model epidemic progression in their geographic region. Interactions between agents are usually accomplished using discrete event simulation due to its advantages. Epidemic progression occurring between time steps is characterized using a system of equations. Geospatial attributes such as population, weather, etc. are incorporated as coefficients in the equations. Each agent uses the same set of epidemic equations associated with a given disease but with different coefficients corresponding to their geography.

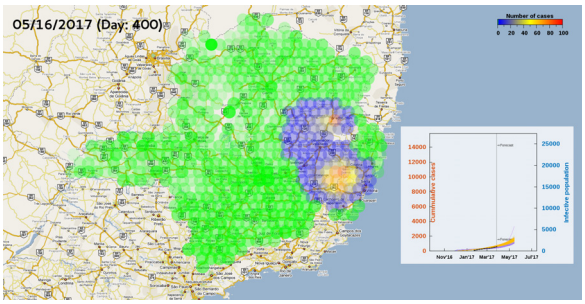


Figure 1: Example of a temporospatial epidemic model with interacting agents represented as circles (Rao et al., 2017)

The epidemic equations are either a system of Ordinary Differential Equations (ODEs) for deterministic simulations or rate-based system for stochastic simulation (SSA). Immaterial of whether deterministic or stochastic methods are used, a conspicuous portion of simulation-runtime is spent on solving the epidemic equations for each agent. The computational time for solving epidemic equations grows polynomially, often to $\gg 90\%$, with changes to different settings, including: decrease in time step size (often 0.01 or 0.005 is used), increase in model size or number of agents, increase in complexity of equations, etc. The computational overheads are further magnified by the need to run 1000s of simulation replications for various analyses. Consequently, reducing runtime of such epidemic simulations is important.

Heterogeneous Computing (HC) involves the use of more than one type of processor for running a program – e.g., a standard CPU and a General Purpose Graphics Processing Unit (GPGPU). A GPGPU is essentially a massive Single Instruction Multiple Data (SIMD) processor. GPGPU is conducive for epidemic simulation because all the agents in a temporospatial model use the same set of equations (*i.e.,* single set of instructions) but with different coefficients (*i.e.,* multiple data). On the other hand, modern CPUs are highly optimized for logic and conditional operations which play a dominant role in discrete event simulations. Consequently, heterogeneous computing performed using CPU and GPGPU holds considerable promise to reduce simulation times.

Motivation & overview of proposed work

Realizing the computational advantages of heterogeneous computing requires considerable technical skill and development of software for different programming paradigms. The issues are compounded by the need to develop both ODE and SSA versions and consistently maintaining disparate versions. Such issues hinder effective use of heterogeneous computing, particularly by the end-users, including epidemiologists and public health experts. Accordingly, to ease the effective use of heterogeneous computing (HC) for epidemic simulations, this paper presents a novel modeling and simulation environment called MUSE-HC and its domain-specific modeling language called Epidemic Description Language (EDL). The paper presents the architecture and design of MUSE-HC along with various verification and performance experiments conducted to assess its overall effectiveness.

RELATED WORKS

The primary focus of this study is modeling and accelerating the simulation of temporospatial epidemic models using Heterogeneous Computing (HC). Several prior investigations have focused on parallel epidemic simulations, including works by Giridharan and Rao (2016), and Rao (2016). These investigations focus on parallel epidemiological simulations using just CPUs. On the other hand, this investigation focuses on using both CPUs and GPGPUs. Consequently, this section compares and contrasts the proposed work with closely related research that involve HC.

Leonenko et al. (2015) propose the use of heterogeneous computing for epidemic simulations. Their work is similar to ours in that discrete event processing is performed on CPU while epidemic propagation is performed on GPGPUs. Their work focus on predefined models written using MATLAB. The use of multiple GPGPUs for epidemic simulation using NVIDIA CUDA is proposed by Shekh et al. (2015). They focus on simulating predefined SEIR individual-based models with contacts arising at buildings or homes. Epidemic simulations using multiple GPGPUs and CUDA in a cluster computing environment has also been investigated by Zou et al. (2013). Their work focuses on simulating predefined contact networks while compensating for communication latencies between the compute nodes on the cluster. Arlindo et al. (2015) explore the use of GPGPUs and CUDA for epidemic simulations. They represent each individual as a string and simulate a predefined epidemic with fixed compartments.

Similar to some of the aforementioned investigations, this work also focuses on simulation of epidemics using heterogeneous computing. However, several novel aspects distinguish this research from prior investigations, namely: ❶ unlike prior investigations that use a fixed model, this study applies to any epidemic model; ❷ a novel, domain-specific modeling language called EDL is proposed in this study, while prior investigations use predefined models; ❸ a key distinction in our study is the use of temporospatial models rather than contact networks used by other investigators; ❹ in contrast to earlier studies that use a hard-coded source code, in this study the CPU and GPGPU source codes are automatically generated from EDL; ❺ in this study we have used OpenCL that is broadly usable and not CUDA that is limited to running only on NVIDIA hardware.

MUSE-HC: ARCHITECTURE & DESIGN

To ease the effective use of heterogeneous computing (HC) for epidemic simulations, we have developed a novel modeling and simulation environment called MUSE-HC, summarized in Figure 2. The primary input is a description of the compartmental model described using a domain-specific, Epidemic Description Language (EDL). EDL has been designed to provide an intuitive, yet expressive constructs for describing epidemic models. The EDL description is parsed into an intermediate Abstract Syntax Tree (AST) using a custom parser. The AST is used by an a code generation module that converts the AST into semantically equivalent versions of source codes for conducting both deterministic and stochastic simulations on CPU and GPGPU. Note that a single EDL description is compiled and transformed to generate both ODE and SSA simulations capable of utilizing heterogeneous compute platforms. A single, intuitive source minimizes overheads associated with model development, validation, and maintenance. The generated model is then complied and linked with the MUSE-HC simulation kernel to produce the final executable. The following subsections provide a more detailed discussion on various aspects of MUSE-HC.

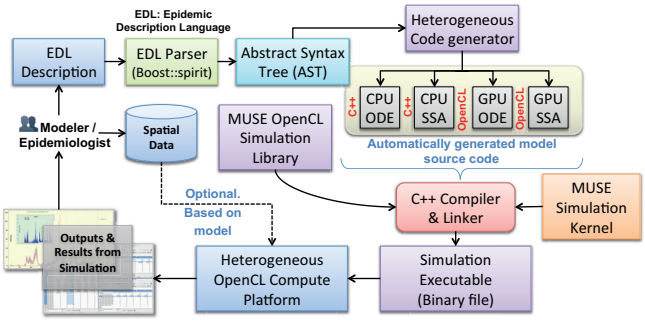


Figure 2: Process of modeling & simulation with MUSE-HC

Epidemic Description Language (EDL)

The Epidemic Description Language (EDL) has been designed to ease effective use of heterogeneous computing (HC) by subject-matter experts, including epidemiologists and public health practitioners. For example, consider the Ebola model shown in Figure 3 that was originally proposed by Legrand et al. (2007). The conceptual model has 6 compartments: Susceptible (S), Exposed (E), Infectious (I), Hospitalized (F), and Funeral (F). Transition of individuals between the compartments is governed by the equations shown with each transition. The equations involve several epidemiological constants that vary for different countries as discussed in detail by Rivers et al. (2014).

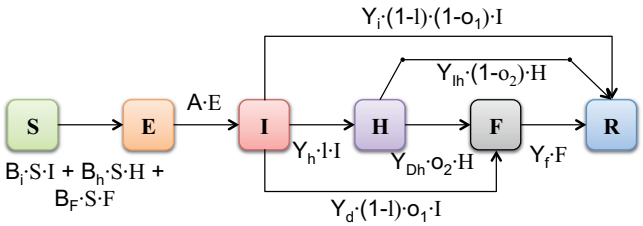


Figure 3: Compartmental model for Ebola proposed by Legrand et al. (2007) and used by Rivers et al. (2014)

The EDL description for the compartmental model in Figure 3 is shown in Listing 1. The description has four code blocks, namely `constants`, `parameters`, `compartments`, and `transitions`. The `constants` block contains named constants used to describe compartment transitions. The `parameters` block contains parameters computed at simulation or run time. These include parameter values that vary with geography, temperature, or other model characteristics. The `compartments` code block lists all of the compartments from the conceptual model.

```
# EDL description for the Ebola model
epidemic Ebola_Liberia {
  # Epidemic constants in this model
  constants {
    Bi = 0.16;    # Contact rate, community
    A  = 0.083;   # Incubation rate (12 days)
    Yi = 0.0667; # Infection duration (15 days)
    # More constants not shown for brevity
  }
  parameters {
     # No time-varying parameters in this model.
  }
  compartments {
     # Comments removed for brevity
     s, e, i, h, f, r;
  }
  transitions {
     e += -1, i += 1 @ (A * e);
     i += -1, h += 1 @ (Yh * l * i);
      # More transitions removed for brevity
  }
}
```

Listing 1: Partial EDL description for model in Figure 3

The last code block in an EDL description defines transitions between the compartments. Each transition consists of two parts, namely: ① a comma-separated list of population changes to a subset of compartments followed by ② the transition rate equation. As illustrated by Figure 3 and Listing 1, EDL syntax has been designed to provide a direct mapping from a conceptual model. Hence EDL is more intuitive for use by epidemiologists and public health experts.

EDL parser and code generator

An EDL description of an epidemic is parsed into an in-memory Abstract Syntax Tree (AST) using the EDL parser as illustrated by Figure 2. The parser has been developed in C++ using `BOOST Spirit` library (version 2.53). The resulting EDL parser produced via `Spirit` is an LL(1) parser that does not require any backtracking due to the straightforward grammar for EDL. The parser performs both syntactic and semantic checks on the input EDL. The output from the EDL parser is an Abstract Syntax Tree (AST). AST is an in-memory data structure that stores the input EDL in a form conducive for further processing.

The AST produced by the EDL parser is used to generate source code for a complete model compliant with MUSE-HC's API. The primary generated artifact is an *agent* class that includes both deterministic and stochastic versions of the model. The deterministic version uses Ordinary Differential Equations (ODEs) that are automatically generated by appropriately combining various transitions for each compartment. The generate source code is in `C` language so that the same code can be run both on the CPU as well as on a GPGPU.

The EDL code generator also produces a top-level simulation (in C++) which provides options for generating a grid of agents in a temporospatial model as shown earlier in Figure 1. The modeler may further enhance or modify the generated source code for fine tuning the simulation as needed. The generated source code is compiled and linked with MUSE-HC to produce the final simulation executable. The resulting executable utilizes the MUSE-HC library to enable simulating the model on heterogeneous computing capable hardware platforms.

The MUSE-HC simulation framework

The heterogeneous computing (HC) capable simulations have been enabled by significantly enhancing a discrete event simulation framework called MUSE. The resulting simulation framework is called MUSE-HC. It has been developed in C++ using its object-oriented capabilities. MUSE-HC also provides a library of algorithms for deterministic and stochastic simulations, including: Runge-Kutta 4th order method, Gillespie's exact Stochastic Simulation Algorithm (SSA), Gillespie's SSA with Tau+Leap optimization, and Mersenne-Twister random number generators. MUSE-HC includes classes that encapsulates the state of an agent. The state includes values for the different compartments in the model along with any time-dependent parameters. The EDL code generator also generates an implementation for the state. The state for each agent is copied to-and-from the GPGPU for heterogeneous computing.

Core simulation and overlapped execution

The salient operations performed by MUSE-HC to run an heterogeneous computing (HC) capable simulation is summarized in Algorithm 1. Upon starting, as part of initialization, MUSE-HC first prepares the OpenCL kernel for GPGPU-based execution. Next, the simulation proceeds in cycles in which each agent processes its next set of events in chronological order. After processing events, each agent indicates if additional HC operation is desired. Agents requesting HC operations are tracked and scheduled for execution in batches as shown in Algorithm 1. The size of each batch is determined by a given `workgroupSize` value. The size of the

workgroup plays an important role in effectively utilizing the computational resources of a GPGPU. Currently, an effective workgroup size is experimentally determined based on the computational capabilities of the CPU versus the GPGPU.

Overlapped execution on CPU & GPGPU

In MUSE-HC, executions on the GPGPU are overlapped with discrete event processing on the CPU as summarized in Algorithm 1. Specifically, the OpenCL kernels are scheduled to execute on the GPGPU in batches and the CPU continues with discrete event processing. Overlapped execution enables amortization of the overheads associated with heterogeneous computing. Using an appropriate workgroup size plays an important role in realizing fast simulations.

```
begin initialization
    // Generate OpenCL kernel for Heterogeneous Computing
end initialization
begin simulation
    Time LVT = 0, GVT = 0, lastLVT = 0
    std::list<HCAgent*> ocl; // HC agents per time step
    while GVT < endTime do
        LVT = scheduler→getNextEventTime();
        if (LVT > lastLVT) or (ocl.size() > workgroupSize) then
            if kernelRunning then // overlapped operations
                wait() // Wait for previous GPU kernel
                copyStateFromGPU();
            end if
            copyStateToGPU(ocl);
            scheduleOpenCLKernel(); // asynchronous launch
            kernelRunning = true; lastLVT = LVT;
        end if
        // Do discrete event processing and track HC agents
        if processNextEvent() then ocl.push_back(agent);
        GVT = updateGVT(); // Update GVT as needed
    end while
end simulation
```

Algorithm 1: Algorithmic overview of salient simulation operations in MUSE-HC

EXPERIMENTS

The first set of experiments focus on verification and validation (V&V) of MUSE-HC. The V&V experiments has been conducted using the extensively validated Ebola model from Figure 3. The experiments has been conducted on a workstation equipped with Intel Xeon E5-2680 v4 CPU (at 2.4GHz with turbo boost to 3.2 GHz) and a NVIDIA Pascal P100 GPGPU. The inset chart in Figure 4 shows the reference model output from Rivers et al. (2014). The larger chart in the figure shows the corresponding `ODE` (deterministic) and `SSA` (stochastic) version of the outputs from MUSE-HC. The outputs from HC simulations were statistically compared to the reference results to establish validity of both deterministic and stochastic simulations. These experiments also verify the models and the simulation infrastructure of MUSE-HC.

In this study, performance assessments of MUSE-HC have been conducted using a synthetic benchmark. The benchmark has been designed to dynamically generate temporspatial models of different sizes, varying number of compartments, and with different interaction patterns. The synthetic models can be used to characterize a broad range of epidemic models in a controlled manner, so as to enable systematic experimental analyses.

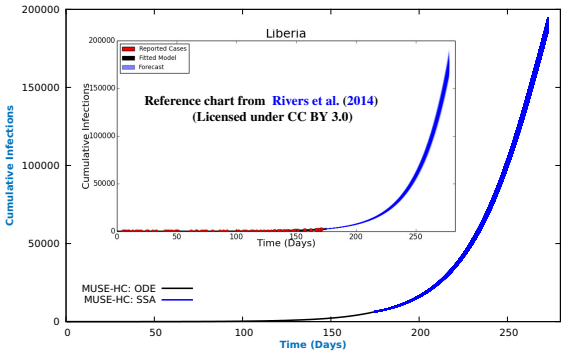


Figure 4: Results from V&V experiments conducted using Ebola model from Rivers et al. (2014) as reference.

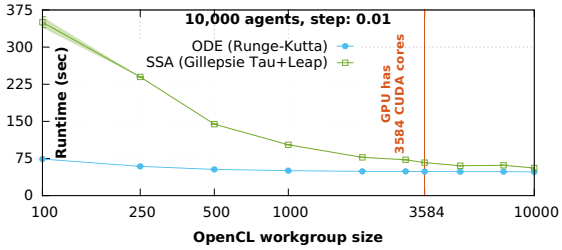
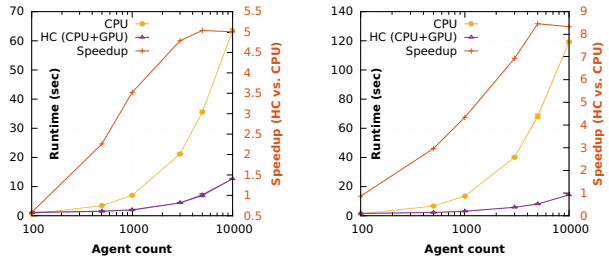


Figure 5: Impact of varying workgroup size

Influence of OpenCL workgroup size: Workgroup size plays an important role in effective use of heterogeneous computing. The chart in Figure 5 illustrates the impact of varying the workgroup size for both deterministic (using Runge-Kutta) and stochastic (Gillespie with Tau+Leap) simulations. The data plotted is the average from 10 independent runs for each workgroup size. The data shows that as the workgroup size increases the runtime decreases because the overheads of copying data from CPU to GPGPU are reduced. However, the improvements tapper off around a workgroup size of 3,000 because the GPGPU has 3,584 SIMD cores and any workgroup larger than that number cannot operate in parallel.

Effect of model size (*i.e.*, number of agents): The charts in Figure 6 illustrates the influence of model size on the runtime. The charts also compare the runtime of the heterogeneous computing (CPU+GPGPU) versus simulation run using 1 core on the CPU. As illustrated by the charts, for small models with just 100 agents, the HC version does not yield significant performance improvement. This is because the compute capabilities of the GPGPU are not fully utilized. With increase in model size, more agents are scheduled to run on the GPGPU thereby better utilizing its compute resources and the HC version outperforms its corresponding CPU-only version. The maximum speedup for deterministic (*i.e.,* ODE-based) version is 5× while the stochastic (*i.e.,* SSA version) is 8.5×. The SSA version yields better per-

formance improvement due to its increased computational needs. In both cases, however, the performance peaks around 4000 agents – the peak is attributed to the maximum of 3,584 SIMD cores available on the GPGPU used in this study.



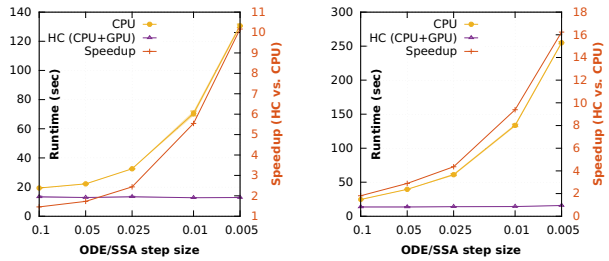
(a) ODE (Runge-Kutta) (b) SSA (Gillespie Tau+Leap)

Figure 6: Impact of varying number of agents (step: 0.01)

Effect of step size: The charts in Figure 7 illustrate the change in runtime as the step size is decreased and computational load increases. As illustrated by the CPU-only curves, for both ODE and SSA versions, the runtime increases polynomially as expected. Nevertheless, the increased computational demands at each simulation time enables more effective use of GPGPU resources and the speed-up realized by the heterogeneous computing (HC) version proportionally increased. With a time step of 0.005, the HC version provides 10× and 16× performance improvement for the ODE and SSA versions respectively, highlighting the overall effectiveness of MUSE-HC.

CONCLUSIONS & FUTURE WORK

Contemporary approaches to epidemic forecasting and containment rely on temporospatial modeling and simulation. Heterogeneous Computing (HC) holds considerable promise to accelerate simulation of temporospatial models. However, HC currently requires considerable technical skill and development of software for different programming paradigms, hindering its use. To ease the effective use of heterogeneous computing (HC) for epidemic simulations, this paper presented a novel modeling and heterogeneous simulation environment called MUSE-HC. MUSE-HC includes a domain-specific modeling language called Epidemic Description Language (EDL) to streamline modeling.



(a) ODE (Runge-Kutta) (b) SSA (Gillespie Tau+Leap)

Figure 7: Impact of varying step size (10,000 agents)

A single EDL description is compiled and transformed to generate both ODE and SSA simulations capable of utilizing heterogeneous compute platforms. The modeling and simulation infrastructure of MUSE-HC has been verified and validated using a well-established Ebola model. The performance assessment of MUSE-HC show that large workgroups sizes are better because it enables faster amortization of the overheads involved in copying data between the CPU's memory and the GPGPU's memory. However, the step size used by numerical methods had the most conspicuous impact on performance improvements realized using HC. With a time step of 0.005, the HC version provides 10× and 16× performance improvement for the ODE and SSA versions respectively, highlighting the overall effectiveness of MUSE-HC and the use of HC for accelerating simulation of large temporospatial epidemic simulations.

ACKNOWLEDGEMENTS

Thanks to Harrison Roth for initial prototype and assessment of MUSE-HC. Support for this work was provided in part by the Ohio Supercomputer Center (Grant: PMIU0110-2).

REFERENCES

Arlindo G.F.; Lauro M.; Jose C.C.; Woerle L.T.; and Anderson S.S., 2015. *CUDA parallel programming for simulation of epidemiological models based on individuals. Mathematical Methods in the Applied Sciences*, 39, no. 3, 405–411.

Giridharan N. and Rao D.M., 2016. *Eliciting Characteristics of H5N1 in High-Risk Regions Using Phylogeography and Phylodynamic Simulations. Computing in Science and Engineering*, 18, no. 4, 11–24. doi:10.1109/MCSE.2016.77.

Legrand J.; Grais R.F.; Boelle P.Y.; Valleron A.J.; and Flahault A., 2007. *Understanding the dynamics of Ebola epidemics. Epidemiology and Infection*, 135, no. 4, 610–621. doi:10.1017/S0950268806007217.

Leonenko V.N.; Pertsev N.V.; and Artzrouni M., 2015. *Using High Performance Algorithms for the Hybrid Simulation of Disease Dynamics on CPU and GPU. Procedia Computer Science*, 51, 150 – 159.

Rao D.M., 2016. *Efficient parallel simulation of spatially-explicit agent-based epidemiological models. Journal of Parallel and Distributed Computing*, 93-94, 102–119. ISSN 0743-7315. doi:10.1016/j.jpdc.2016.04.004.

Rao D.M.; Asbun C.; and Stamper P.D., 2017. *Forecasting and Assessment of Autochthonous Yellow Fever Outbreak in Brazil*. In *66th Annual Meeting*. American Society for Tropical Medicine and Hygiene, Baltimore Convention Center, Baltimore, Maryland, USA. doi:10.13140/RG.2.2.18785.40805. Poster.

Rivers C.M.; Lofgren E.T.; Marathe M.; Eubank S.; and Lewis B.L., 2014. *Modeling the Impact of Interventions on an Epidemic of Ebola in Sierra Leone and Liberia. PLOS Current Outbreaks.*

Shekh B.; Doncker E.D.; and Prieto D., 2015. *Hybrid multithreaded simulation of agent-based pandemic modeling using multiple GPUs*. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 1478–1485.

Zou P.; shuai L Y.; da Wu L.; li Chen L.; and ping Yao Y., 2013. *Epidemic simulation of a large-scale social contact network on GPU clusters. SIMULATION*, 89, no. 10, 1154–1172.

SIMULATION IN ECONOMICS AND FINANCE

VOLATILITY REGIME ANALYSIS OF BITCOIN PRICE DYNAMICS USING MARKOV SWITCHING GARCH MODELS

Anna Bonello, David Suda
Department of Statistics and Operations Research, University of Malta
emails: anna.bonello.14@um.edu.mt, david.suda@um.edu.mt

KEYWORDS

Finance, Forecasting, Estimation, Filtering, Time Series Analysis

ABSTRACT

One of many applications of machine learning is that of correctly classifying market and economic phases. A popular approach to this is using Markov switching models. This paper will look at the implementation of Markov switching generalised autoregressive conditional heteroscedastic (MSGARCH) models with normal and t-distributed innovations to recent Bitcoin/US Dollar price dynamics, and also show that these can be an improvement over single-regime models of the same kind, by demarcating high and low volatility regimes. Furthermore, we also look at both maximum likelihood estimation, and Bayesian estimation via Markov Chain Monte Carlo. The predictive performance is also analysed using risk management tools such as value-at-risk and expected shortfall, and we show that the two-regime model with t-distributed innovations provides the best fit.

INTRODUCTION

In recent history, cryptocurrencies have become a global phenomenon which, thanks to its blockchain technology, is challenging to change the world of finance and beyond. Essentially, a cryptocurrency is a digital currency with the main purpose of decentralising control from the central banking system. Bitcoin, with its inception in 2009, was the initial cryptocurrency. However, since then, other cryptocurrencies have been introduced (a number of which also perished). Last year, the stock market was the least volatile in the last decade. However, this was not the case in regard to Bitcoin and other cryptocurrency prices. The first cause for this volatility is that cryptocurrency wealth distribution is more disproportionate than that of traditional financial assets. The second cause is that public understanding is quite divided and subjective, and thus will make the price swing even more. Finally, regulation plays a central role in cryptocurrencies' exchange rates' intense volatility. Even though cryptocurrencies run on a decentralised banking system, various governments have made attempts to regulate these assets. In 2015, the U.S. Commodity Futures Commission declared that cryptocurrencies essentially are not considered as currencies but more as a commodity and hence could not be regulated. Recently however, governments and other entities have been taking a more hostile approach. In September 2017, China's government ceased domestic exchanges while in January 2018, the South Korean Government also shut down anonymous cryptocurrency trading. More recently, there has also been a ban on cyrptocurrency advertising from the likes of Facebook and Google. Consequently, the 'bull' year cryptocurrencies had in 2017 on the back of the hype surrounding them was followed by a 'bear' first half of 2018, which saw them crash to much lower levels than their 2017 peaks.

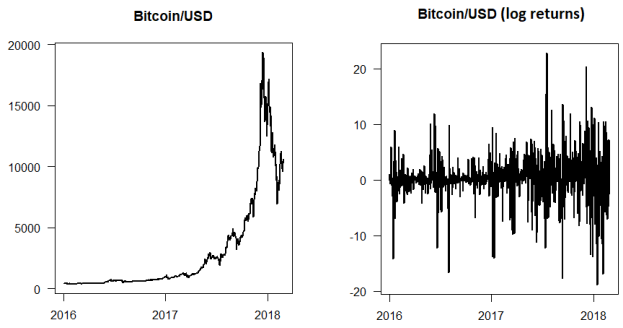


Figure 1: Sequence charts for Bitcoin/US Dollar (left) and their log returns (right) for the period 1/1/2016-28/2/2018.

Modelling literature on cryptocurrencies is still quite limited, though the theme has been generating some interest lately. Chan et al. (2017) fit various parametric distributions on cryptocurrency returns. Furthermore, Bouri et al. (2017), Chu et al. (2017), Katsiampa (2017), Baur et al. (2018) and Stavroyiannis (2018) fit generalised autoregressive conditional heteroscedastic (GARCH) models and its variants in their single-regime form. Kodama et al. (2017), on the other hand, look at the application of Markov switching autoregressive models to Bitcoin. Currently, a preprint by Ardia et al. (2018) can be found on SSRN that looks at the implementation of symmetric and asymmetric models of the MSGARCH type using a Bayesian approach to Bit-

coin/US Dollar. In contrast to Ardia et al. (2018), in this paper we look solely at MSGARCH models of the symmetric type on Bitcoin/US Dollar, but from both a frequentist and Bayesian perspective. Another difference is that we do not take the whole history but concentrate on a recent 26 month dataset of 790 daily readings (1/1/2016-28/1/2018) - see Figure 1. The intention is that of looking at more homogenous dynamics, hence not allowing estimation of the model to be affected by initial periods when Bitcoin was worth very little and volatility was low.

MODEL DEFINITION

There is more than one Markov switching approach to GARCH modelling (see e.g. Gray 1996, Klaassen 2002). We focus on the approach by Haas et al. (2004). This model is characterised by,

$$\epsilon_n = Z_n \sigma_{\Delta_n, n} \,,$$

for $n = 1, 2, ...$ where (ϵ_n) is a time series of residuals, (Z_n) is a sequence of normal i.i.d random variables with mean zero and unit variance, the $k \times 1$ vector $\boldsymbol{\sigma}_n^2 = [\sigma_{1,n}^2, \sigma_{2,n}^2, \ldots, \sigma_{k,n}^2]'$ of regime variances follows the multivariate GARCH(1,1) equation,

$$\boldsymbol{\sigma}_n^2 = \boldsymbol{\alpha}_0 + \boldsymbol{\alpha}_1 \epsilon_{n-1}^2 + \boldsymbol{\beta} \boldsymbol{\sigma}_{n-1}^2 \,,$$

where $\boldsymbol{\alpha}_i = [\alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,k}]'$, $i = 0, 1$; $\boldsymbol{\beta}$ =diag$\{\beta_1, \beta_2, \ldots, \beta_k\}$, (Δ_n) is a Markov chain with finite state space $S = \{1, 2, \ldots, k\}$ and an irreducible and primitive $k \times k$ transition matrix $\mathbf{P}$, and (Z_n) and (Δ_n) are independent. The inequalities $\alpha_{0,j} > 0$, $\alpha_{1,j} \geq 0$ and $\beta_j \geq 0 \,, \forall j = 1, \ldots k$ are required to guarantee positivity of the conditional variance. In our case, the ϵ_n's are the log returns (which were found to be uncorrelated). For comparison, we shall also consider the single-regime GARCH by Bollerslev (1986).

In this paper we consider two estimation approaches. One is maximum likelihood estimation (MLE), which is performed using a recursive filter for Markov switching models developed by Hamilton (1989). The other is the Bayesian approach, via the MSGARCH R package developed by Ardia et al. (2016), which uses the robust adaptive Metropolis-Hastings approach introduced by Vihola (2012) to simulate from the posterior distribution. Discussions on moment conditions, volatility persistence and regime persistence for the MSGARCH model can be found in Haas et al.(2004). Volatility persistence in a particular regime is determined by how much $\alpha_{1j} + \beta_j$ is close to 1 for $j = 1, ..., k$ (the closer it is to 1, the more persistent it is). It represents the persistence of the effect of large shocks on future volatility. Regime persistence, on the other hand, is determined by the diagonals of the transition matrix $\mathbf{P}$. Moment conditions are also taken care of by the MSGARCH package. Also, theory on regime inference can be found in Klaasen (2004) Appendix D, which also extends to the Haas model. This will be crucial in determining the high and low volatility market phases.

Table 1: AIC and BIC values for the four MLE model fits

	$N1$	$NMSG2$	$t1$	$tMSG2$
AIC	4203.31	4019.32	4030.08	**4001.42**
BIC	4217.32	4056.70	4048.77	**4043.47**

Table 2: MLE estimates and corresponding standard errors for parameters of the single-regime models

	$N1$		$t1$	
	Estimate	Std. Err.	Estimate	Std. Err.
α_0	0.2669	0.0025	0.1040	0.0019
α_1	0.1834	0.0035	0.1635	0.0038
β_1	0.8145	0.0000	0.8302	0.0002
ν	-	-	3.6730	0.0100

MODEL ESTIMATION AND REGIME INFERENCE

We shall fit four different models on this dataset and use the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) to determine the best model fit. These are the single-regime GARCH model with normal and t-distributed innovations, and their two-regime counterparts. We shall refer to these models as $N1$, $t1$, $NMSG2$ and $tMSG2$ respectively. We do not present models for higher regime orders as these did not yield significant improvements over the two-regime case. Among the single-regime models, the t-distributed version performs the best. AIC and BIC both agree that the best model is $tMSG2$. Further information can be seen in Table 1.

In Tables 2 and 3 one can find the model fits for both the single-regime GARCH models and their two-regime

Table 3: MLE estimates and corresponding standard errors for parameters of the two-regime models

	$NMSG2$		$tMSG2$	
	Estimate	Std. Err.	Estimate	Std. Err.
α_{01}	0.0173	0.0004	0.0948	0.0025
α_{11}	0.0800	0.0010	0.0555	0.0018
β_1	0.8208	0.0007	0.8780	0.0019
α_{02}	4.2402	0.0675	2.7089	0.0700
α_{12}	0.2098	0.0074	0.1161	0.0030
β_2	0.7864	0.0002	0.8640	0.0028
ν	-	-	3.3042	0.0235
p_{11}	0.7775	0.0024	0.9762	0.0006
p_{21}	0.5907	0.0030	0.0231	0.0006

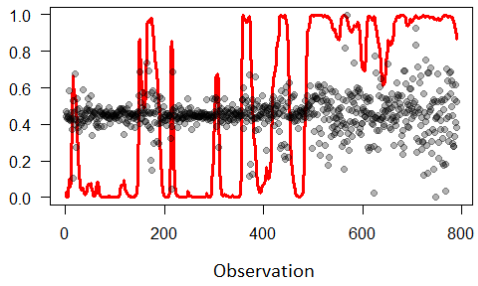


Figure 2: Smoothed probabilities of being in regime 2 (continuous line) superimposed on the model residuals (black, dotted) using MLE for $tMSG2$.

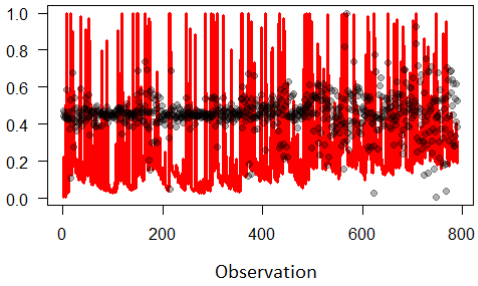


Figure 3: Smoothed probabilities of being in regime 2 (continuous line) superimposed on the model residuals (black, dotted) using MLE for $NMSG2$.

analogues. From Table 3, it can be deduced from the probability estimates that the stationary probabilities for $NMSG2$ are given by 0.73 and 0.27 for the first and second regime respectively, while in the case of $tMSG2$ these are given by 0.49 and 0.51. Thus, these two models do not agree on which regime is the most common, as tMSG2 gives almost equal probabilities to the two regimes. We note that because of the fat tails of the Student's t-distribution, the regimes in tMSG2 are highly persistent with estimated staying probabilities for p_{11} and p_{22} larger than 0.95. The parameters show that the two regimes are quite different and that the first regime is related to more calm market conditions, while the second regime is related to more rough market conditions.

Table 4: Bayesian estimation model fit (DIC)

	$N1$	$NMSG2$	$t1$	$tMSG2$
DIC	4206.85	4030.44	4492.37	**4004.44**

We shall now look at the smoothed regime probabilities for the two-regime models. Figure 2 displays the smoothed probabilities of being in the regime related to rough market conditions, that is the second regime, when considering t-distributed innovations. In comparison to Figure 1, it can be seen that, as soon as we have higher log return fluctuations, especially in recent periods, the MSGARCH model assigns higher probabilities to the second regime. Data corresponding to observations from October 2017 up to the end of the series all have smoothed probability of being in the second regime higher than 0.5. The reason is that at the end of 2017, Bitcoin prices changed constantly with drastic movements, thus high volatility was present. Figure 3 shows the smoothed regime probabilities if we were to assume normally distributed innovations. In this case, we have numerous regime shifts along the dataset, since in this case, large shifts in the log returns are automatically attributed to a regime change.

Table 5: Bayesian estimates (empirical means of simulated posterior values) and corresponding standard deviations for parameters of the single-regime models.

	$N1$		$t1$	
	Mean	Std. Dev.	Mean	Std. Dev.
α_0	0.3189	0.0941	0.1461	0.0694
α_1	0.1885	0.0246	0.1719	0.0256
β_1	0.8067	0.0256	0.8180	0.0261
ν	-	-	3.7539	0.3242

Table 6: Bayesian estimates (empirical means of simulated posterior values) and corresponding standard deviations for parameters of the two-regime models

	$NMSG2$		$tMSG2$	
	Mean	Std. Dev.	Mean	Std. Dev.
α_{01}	0.0141	0.0006	0.0410	0.0451
α_{11}	0.1007	0.0259	0.1494	0.0348
β_1	0.8096	0.0292	0.8153	0.0320
α_{02}	16.9849	12.3391	30.0384	15.0252
α_{12}	0.3217	0.1591	0.3449	0.2192
β_2	0.5240	0.2277	0.2792	0.2135
ν	-	-	4.6389	1.2309
p_{11}	0.8113	0.0456	0.8943	0.0472
p_{21}	0.6572	0.1168	0.5301	0.2400

We now move on to estimating the model in a Bayesian way using (Monte Carlo Markov Chain) MCMC, where in this case we treat the parameters as random. In the prior, we treat parameters as independent and assume diffuse truncated priors on each, then we use robust adaptive Metropolis-Hastings to sample from the posterior. Following a burn-in period, we take 50,000 draws with thinning of 200 iterations in order to reduce autocorrelation. In this case, we shall use the deviance information criterion (DIC) for comparison. Table 4 confirms that the best model is the two-regime with Student's t-distribution as it has the lowest DIC. Once again, the three-regime model did not show any improvements and related outputs will not be presented.

Starting with the single-regime models we note that, except for ν, all the parameters have small standard deviations and standard errors. Comparing the Bayesian esti-

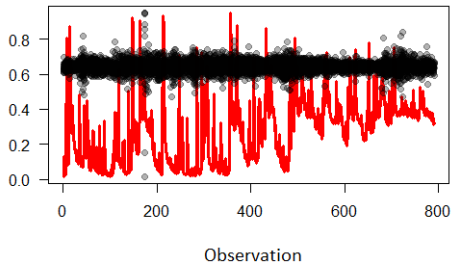


Figure 4: Smoothed probabilities of being in regime 2 (continuous line) superimposed on the model residuals (black, dotted) using Bayesian estimation for $tMSG2$.

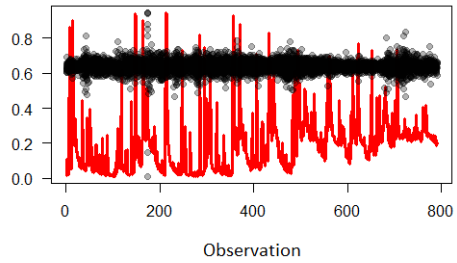


Figure 5: Smoothed probabilities of being in regime 2 (continuous line) superimposed on the model residuals (black, dotted) using Bayesian estimation for $NMSG2$.

mates obtained via the mean in Table 5 to those derived through MLE in Table 2, we note that the Bayesian estimates are very similar to the MLE ones. If we look at the parameters of the two-regime models, we note that for both models, the parameter α_{02} has a very high standard deviation, which means that the values in the sample vary from one another. Also, the mean value from the posterior sample is considerably high, and hence this Bayesian estimate will considerably effect the volatility in the second regime. As a result of this high Bayesian estimate, the volatility persistence in the second regime calculated by $\alpha_{12}+\beta_2$ has decreased considerably. Furthermore, through Bayesian estimation, the persistence found in Bayesian estimates of the staying probabilities p_{11} and p_{22} of the $tMSG2$ model has also been lowered. This may warrant the investigation of the use of informative priors, but such possible improvements will not be investigated here. The stationary probabilities provided by $NMSG2$ are given by 0.78 and 0.22 while those provided by $tMSG2$ are given by 0.83 and 0.17 for the first and second regime respectively. Hence, now both models agree that the first regime is more common. Figures 4 and 5 give the smoothed probabilities of being in the second regime from Bayesian estimation, and it can be seen that in this case, both models have numerous regime shifts, and for $tMSG2$, the chance of persisting in the high volatility regime is decreased.

FORECASTING PERFORMANCE

Generally, practitioners are more interested in high forecast accuracy than goodness of fit. The main aim of this section is that of analysing whether the additional complexity of a Markov switching framework is useful, and whether using a Bayesian framework with diffuse priors leads to better risk prediction. We shall backtest to examine the predictive performance of the estimated models where the accuracy of the predictions is analysed using value-at-risk (VaR).

We shall examine model prediction performance by carrying out historical backtesting. Backtesting is a technique in which an investigation is run in order to determine the predictive performance of several models, based on historical data. It can be done by either using a rolling window method or an expanding window method. We shall use the expanding window method. In this case, we divide our time series into the in-sample observations $\epsilon_1,...,\epsilon_{N-h}$ and the out-of-sample observations $\epsilon_{N-h+1},...,\epsilon_N$, To obtain the VaR for each ϵ_{N-h+i}, we use all possible prior information (in the rolling window case, we use a fixed window length of information) on the predictive density. We denote the predictive density for the frequentist case by $f_{N-h+i}\left(\epsilon_N-h+i+1|\boldsymbol{\theta}\right)$ where $\boldsymbol{\theta}$ denotes the vector of parameters of the model. In the Bayesian case, we use all the simulated parameters $\boldsymbol{\theta}^{(m)}$ for $m=1,...,M$ obtained from MCMC (in our case $M=50,000$), and obtain the predictive density approximation $\hat{f}\left(\epsilon_{N-h+i+1}\right)=\frac{1}{M}\sum_{m=1}^{M}f_{N-h+i}\left(\epsilon_{N-h+i+1}|\boldsymbol{\theta}^{(m)}\right)$. To calculate the VaR of $\epsilon_{N-h+i+1}$ at level τ given $\epsilon_1,...,\epsilon_{N-h+i}$, which we denote by $VaR^{\tau}_{N-h+i}\left(\epsilon_{N+1}\right)$, we take

$$VaR^{\tau}_{N-h+i}\left(\epsilon_{N+1}\right)=-q^{\tau}\left(\epsilon_{N+1}\right),$$

which is the negative of the only τ-quantile of ϵ_{N+h-i} given the predictive density.

Table 7: p-values for Kupiec and Christoffersen tests for both maximum likelihood and Bayesian estimation AE denotes the actual over expected exceedance ratio.

	AE	Kupiec	Christoffersen
	Maximum likelihood estimation		
$N1$	1.6842	0.0478	0.1196
$NMSG2$	1.7895	0.0239	0.0719
$t1$	1.7895	0.0239	0.0719
$tMSG2$	1.4737	0.1604	0.2436
	Bayesian estimation		
$N1$	1.6842	0.0478	0.1196
$NMSG2$	1.6842	0.0478	0.1196
$t1$	1.7895	0.0239	0.0719
$tMSG2$	1.6842	0.0478	0.1196

In our case we shall take the number of out-of-sample ob-

servations to be equal to 190, the number of in-sample observations equal to 600, and re-estimate the model every 20 observations. We then use the Kupiec and Christoffersen tests (see Kupiec 1995 and Christoffersen 1998 for more detail). In the Kupiec test, given x out of h breaches for the VaR, we test the null hypothesis that the breach rate is τ. We are therefore testing whether the proportion of breaches $\frac{x}{h}$ is significantly different from the τ level stipulated by the VaR. In the Christoffersen test, we denote by κ_0 the probability of a breach occurring given no previous breach, and by κ_1 the probability of a breach occurring given a previous breach. Both quantities can be estimated from the out-of-sample observations and the null hypothesis tests whether $\kappa_0 = \kappa_1$. In this case, in the null hypothesis we are testing whether breaches are occurring independently, in conjunction with the hypothesis that the breach rate is τ. Table 7 yields the outputs for the Kupiec and Christoffersen tests. We see that at the 0.05 level, all models accept the null hypothesis of independence stipulated by the Christoffersen test, both when the frequentist approach and the Bayesian approach are used. However, the null hypothesis for the Kupiec test is only accepted at the 0.05 level for $tMSG2$ when maximum likelihood estimation is applied, hence providing the best VaR-forecasting performance.

Finally, we also backtest whether $tMSG2$ estimated via MLE, which is the best performing model for the one-step VaR, is good for forecasting multistep VaR and explaining tail behaviour. We do the former by similarly applying the Kupiec and Christoffersen tests on the 5-step VaR. This yields p-values of 0.1998 and 0.3794 respectively, accepting as desired the null hypothesis in both cases. On the other hand, we do the latter by looking at a test based on expected shortfall (ES), where to calculate the ES of $\epsilon_{N-h+i+1}$ at level τ given $\epsilon_1, ..., \epsilon_{N-h+i}$, which we denote by $ES^{\tau}_{N-h+i}(\epsilon_{N-h+i+1}])$, we take

$$ES^{\tau}_{N-h+i}(\epsilon_{N-h+i+1}]) = -\tau^{-1}E[\epsilon_{N-h+i+1}1_A],$$

provided the innovations are continuously distributed, where A represents the set of all possible outcomes such that $\epsilon_{N-h+i+1} \leq q^{\tau}(\epsilon_{N-h+i+1})$ and 1_A the corresponding indicator function. For this, we use a test proposed by McNeil and Frey (2002). In this test, excess loss (EL) is defined by

$$EL = \left(\epsilon_{N-h+i+1} - ES^{\tau}_{N-h+i}(\epsilon_{N-h+i+1})\right) \times 1_{\epsilon_{N-h+i+1} \leq q^{\tau}(\epsilon_{N-h+i+1})}.$$

In the null hypothesis, we test $E[EL] = 0$ via a bootstrap-type test. $tMSG2$ with maximum likelihood estimates yields a p-value of 0.3187 for this model, which means that the model is also suitable to model the innovations at the tails.

CONCLUSION

In this paper we applied MSGARCH models with two regimes, and the single-regime counterpart for comparison, on Bitcoin/US Dollar using both normal and t innovations. In the frequentist approach, it was found that the $tMSG2$ was the best of the models considered where goodness-of-fit is concerned. It was also found to have the best risk forecasting performance for one-step VaR, and also performed well for multistep VaR and in explaining tail behaviour. This conclusion partly corroborates the current pre-print by Ardia et al. (2018) which picks two-regime models as being superior to single-regime ones. We also considered Bayesian estimation with diffuse priors as presented in Ardia et al. (2016). Alas these gave an inferior forecasting performance and the regime persistence present in the maximum likelihood version for $tMSG2$ was lost. One must, however, not exclude the Bayesian approach outright and one may consider expert or empirical Bayes priors, or alternative Bayesian estimates, to check whether these yield an improvement. With the novelty and short history of the asset in question, however, using the Bayesian approach effectively may prove to be a challenge.

This study has, on the whole, been an interesting study in determining the different volatility regimes of the Bitcoin/US Dollar series. Nonetheless, we feel that the ever changing landscape of cryptocurrencies makes it premature, as yet, to use this type of volatility analysis as a reliable forecast of risk. As has been mentioned, the MSGARCH model in this paper is strongly linked with hidden Markov and related models, and aside from inferring the different volatility states such an asset goes through, trying to determine whether such models can be used to identify 'bull' and 'bear' market phases would also be interesting approach to consider in the future. Indeed, volatility and bull/bear behaviour are intricately linked, and one could potentially use the former to infer the latter.

REFERENCES

Ardia, D., Bluteau, K., Boudt, K., Catania, L. and Trottier, D.A. (2016). Markov switching GARCH models in R: The MSGARCH package. CRAN.

Ardia, D., Bluteau, K. and Ruede, M. (2018). Regime changes in Bitcoin GARCH volatility dynamics. SSRN Preprint.

Baur, D.G., Dimpfl, T. and Kuck, K. (2018) Bitcoin, gold and the dollar - a replication and extension. Financ. Res. Lett., 25:103-110.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. J. Econometrics, 31(3):307327.

Bouri, E., Azzi, G. and Dyhrberg, A.H. (2018). On the return-volatility relationship in the Bitcoin market around the price crash of 2013. Econstor Economics

Discussion Papers, 2016-41.
Christoffersen, P.F. (1998). Evaluating Interval Forecasts. Int. Econ. Rev., 39:841-862.
Chan, S., Chu, J., Nadarajah, S. amd Osterrieder, J. (2017). A statistical analysis of cryptocurrencies. Journal of Risk and Financial Management, 10(12):1-23.
Chu, J., Chan, S., Nadarajah, S. and Osterrieder, J. (2017). GARCH modelling of cyprtocurrencies. Journal of Risk and Financial Management, 10(17):1-15.
Gray, S.F. (1996). Modeling the conditional distribution of interest rates as a regime-switching process. J. Financ. Econ., 42(1):27-62.
Haas, M., Mittnik, S. and Paolella, M.S. (2004). A new approach to Markov switching GARCH models. J. Financ. Econ., 2(4):493-530.
Hamilton, J.D. (1989). A new approach to economic analysis of nonstationary time series and the business cycle. Econometrica, 57:357-384.
Katsiampa, P. (2017). Volatility estimation for Bitcoin, a comparison of GARCH models. Economic Letters 158:3-6.
Klaasen, F. (2002). Improving GARCH volatility forecasts with regime-switching GARCH. Empir. Econ., 27(2):363394.
Kodama, O., Pichl, L. and Kaizoji, T. (2017). Regime change and trend prediction for Bitcoin time series data. Proceedings of the CBU International Conference on Innovations in Science and Education 5:384-388.
Kupiec, P. (1995). Techniques for verifying the accuracy of risk measurement models. J. Deriv., 3(2):73-84.
McNeil, A. J. and Frey, R. (2000). Estimation of tail-related risk measures for heteroscedastic financial time series: an extreme value approach. J. Empir. Financ., 7(3-4):271300.
Stavroyiannis, S. (2018). Value-at-risk and related measures for the Bitcoin. Journal of Risk Finance 19(2):127-136.
Vihola, M. (2012). Robust adaptive Metropolis algorithm with coerced acceptance rate. Stat. Comput., 22(5):997-1008.

MODELS AS PART OF THE WELFARE SYSTEM AS PART OF THE ECONOMIC MEASURES FOR THE STATE OF CRISIS IN BUILDING RESILIENCE

Assoc. Prof.Jozef Ristvej, Ph.D.
Department of Crisis Management
University of Žilina, www.uniza.sk
Univerzitná 1, 01026 Žilina, SK

Roman Ondrejka, Ph.D.
Transport Research Institute, Inc.
R&D Division, www.vud.sk
Veľký Diel 3323, 010 08 Žilina, SK

professor Ladislav Šimák, Ph.D., Michal Titko, Ph.D.
MSc. Michaela Jánošíková
Department of Crisis Management, University of Žilina
Univerzitná 1, 01026 Žilina
Slovakia
E-mail: Jozef.Ristvej@fbi.uniza.sk

KEYWORDS
Crisis Management, Simulation, Decision Support.

ABSTRACT

The assessment of the current state of the citizens' social welfare in crisis situations and use of simulations is challenge of today crisis management, especially in support of decision making, as well as the comprehensive evaluation of the risks threatening in the process of providing the overall support for the citizens in crisis situations. In this paper we deal with welfare system as part of the economic measures for the state of crisis in the Slovak Republic and also the position and tasks of the centres of the social welfare for the citizens in crisis situations at the level of the local state administration is analysed. One of the main task of our team is to develop models and scenarios that can be used as a supporting materials and background for exercises within the constructive simulation in crisis management. The team use this approach in specialized Laboratory of Modelling and Simulation of Crisis Phenomena for Improving Efficiency of Decision Making at the University Science Park of the University of Žilina.

INTRODUCTION

The security of the Slovak Republic is based on the political, economic, humanistic, legal, social, cultural, historical, military and strategic, internal security and ecological elements and their mutual links. It represents a structured multidimensional system that protects:

- the lives, freedom and property of the citizens,
- the society with its intellectual and material values,
- the state as a whole (unit).

The achieved level of the state's security is very closely connected with the real needs of the society which wants to feel safely with the current and potential threats on the one hand and with the international reputation, economic opportunities, technological prerequisites, human potential and a whole range of other assumptions on the other hand. The security system of the state represents the institutional and functional expression of the state's security policy. It is a system of the public administration's institutions, forces and means of the military forces, armed security units, emergency units and services, corporations and self-employed persons, the appropriate legal standards, mutual links and relations through which the security of the state, its citizens, material values and environment are ensured. An appropriately working security system is thus an inevitable assumption for achieving the state's overall security.

SECURITY SYSTEM OF THE SLOVAK REPUBLIC

The security environment from the global point of view but also from the viewpoint of every country or region changes and develops continuously. It is modified by a whole range of external as well as internal factors out of which majority is of a stochastic character; however, some of them have a strictly deterministic nature. A comprehensive analysis of the security environment has always a quasi-objective character and is limited by a range of subjective attitudes and the rate of accessibility and truthfulness of the input data.

The appropriate bodies of the legislative and executive powers, prosecution, local state administration, territorial self-government as well as the corporations and self-employed persons have responsibility for guaranteeing the security of the state in the range of determined scope of authority and in the framework of the stated competencies. The security state of the country is created by sub-systems, management areas of the security system, foreign security policy of the state, by providing the defence of the country, citizens' protection, protection of the property and the environment, the internal security and order in the country as well as the area of ensuring the forces, means and resources inevitable for the activities of the security system comprehensively.

The area of the Slovakia's security system is created by bodies of the public power of the state and the corporations and self-employed persons which continually ensure planning, organising, coordinating, realising and checking the decisive measures for ensuring the constant functionality of the security system. The Slovak government and its Ministry of Defence and Ministry of Interior play the most important role here. It is linked with the area of the foreign security policy created by the bodies of the state's public power, especially the Ministry of

Foreign and European Affairs of the Slovak Republic, the Ministry of Defence, Ministry of Interior and Ministry of Economy of the Slovak Republic. They make decisions and carry out activities in the area of forming a stable external security environment. The area of defence is created by the state's public power bodies and the corporations and self-employed persons that create assumptions for ensuring the state's defence, preparation for the state's defence and for fulfilment of the obligations resulting from the international treaties concerning the common defence through their activities and measures. The area of protecting the citizens, property and environment as well as the internal security and order in the country is formed by the bodies of the state power and appropriate corporations and self-employed persons whose activities and measures ensure prevention and solve the threats of the non-military character.

The area of ensuring the necessary prerequisites is the last sphere of the security system. It is created by the state's public power bodies and corporations and self-employed persons participating in ensuring the material and technical prerequisites. Their activities and measures form assumptions for ensuring the functionality of the individual sections of the security system. One part of this area is e.g. the System of Economic Mobilisation, the state material reserves, but also the critical infrastructure elements. The result of their activities is especially the summary of the organisational, economic, material and technical and financial measures without which it would not be possible to solve any major crisis situations.

ECONOMIC MEASURES FOR THE STATES OF CRISIS

The common crisis events can escalate into the State of Crisis (SoC) which we perceive as the crisis situations. In the law of the Slovak Republic (SR) there are 4 SoCs with corresponding examples of the crisis situations: State of Emergency (SoEm) (e.g. severe floods, severe earthquakes, major industrial accidents, etc.), State of Exception (SoEx) (e.g. riots, terrorism, etc.), State of Ware (SoW) (e.g. the threat of military aggression, threat of the war declaration, etc.) and War (e.g. war outside or inside the SR borders) (Hudák et al. 2013). The SoCs are legal states declared by the government of the SR and characterised by the fact that their solutions are no longer sufficient by the common forces and resources of the integrated rescue system. In order to deal with the SoC specific measures are used (economic mobilisation, state material reserves, directive public restrictions, using of armed forces, etc.) which can vary depending on particular CS solution requirements. For references see (Act No. 179/2011) and (Decree 2011).

The economic mobilisation (EM) is one of the basic pillars of solving the crisis situations, transformation of economy, ensuring the basic needs inevitable for surviving the inhabitants and for ensuring the activities of the armed forces, armed security units, emergency units of the integrated rescue system and subjects of the economic mobilisation.

The EM system fulfils a whole range of measures which belong to the competencies of the economic mobilisation. The particular EM subjects are established according to the law or by decisions of the competent body. They represent especially individual bodies of the state administration, self-government (72 district offices, 2,930 villages and towns, 8 regional territorial units) and defined entrepreneurial subjects (also budget and state-funded institutions) which will carry out specific activities and measures (several hundreds of subjects) during the state of crisis in the area of production, trade, services, finance, social welfare and healthcare and realise specific activities and measures in selected spheres of the out-of-economy social superstructure and other institutions (Slovak Radio and TV, National Bank of Slovakia, the Treasury, Social Insurance and public universities). Those subject ranked in the EM are preparing for realising particular EM measures already during the state of security and carry them out after declaring some of the states of crisis or based on an order of the competent body.

The EM measures ensure fulfilling the basic functions of the state during the states of crisis and their financing is carried out by the state from the financial means of the state budget. The EM measures can be divided to the general and specific ones. The general measures are fulfilled by all EM subjects. The specific measures are fulfilled only by those subjects whose duty results from their position in the crisis management system or their tasks determined by the decision of the competent body.

The Ministry of Labour, Social Affairs and Family of the Slovak Republic activates and creates the Centre of Concentrated Social Welfare (CCSW) for ensuring this comprehensive social assistance. The CCSW is a place of concentrated activities of institutions providing the social welfare during the crisis situations. The decisive role is a concentrated, fast and proactive assistance for the affected citizens and ensuring the social stability. The CCSW is the basic element for solving the social welfare during the crisis situation which is established in the affected regions as close as possible to the hit persons. The determined subjects working in the framework of the CCSW organise their activities in such a way that the persons hit by the consequences of the crisis situation can receive all relevant things resulting from the law within 30 days from the first contact with them.

The Ministry of Labour, Social Affairs and Family of the Slovak Republic, the Centre of Labour, Social Affairs and Family, the labour, social affairs and family offices, the National Inspectorate of Labour and inspectorates of labour, Social Insurance Company, Institute for Working Rehabilitation of Disabled Citizens, Institute for Labour Recovery of Citizens, Rehabilitation Centre for Blind and Partially Sighted Persons, regional territorial units, towns and villages, legal entities and self-employed persons and NGOs dealing with activities in the area of social welfare (Methodological Guideline No. 16053/2014) belong to the subjects participating in the CCSW activities.

After activating the CCSW the obligation of the labour, social affairs and family offices, the affiliations of the Social Insurance Company, the National Inspectorate of Labour and the regional territorial unit and other institutions have to provide employees for its needs according to their territorial scope. After activating the CCSW the municipality provides space for its activities free of charge and during the crisis situation the natural persons are obliged to submit to the social welfare organisation.

The possibility to establish the CCSW is a relatively new measure in the framework of the social welfare system for

the citizens' needs during crisis situations. Its effective work depends on the preparedness of the competent employees as well as on the quality of the crisis plans. Several exercises have been carried out for verifying the functionality and preparedness of the CCSW. In the framework of them the CCSW activities were trained and subsequently adapted or adjusted for the needs of practice through identifying the problems. There were realised exercises into which the CCSW were also implemented.
One of the primary requirements of the current crisis managers for solving the aforementioned situations (but also others) is to have an access to the actual data from any place (as mentioned before the CCSWs are field workplaces) and the possibility to exchange the information between the involved subjects quickly to ensure adequate help for the affected inhabitants in the shortest possible time. At the same time the solution of such situations requires a comprehensive planning of tasks, resources and procedures in all spheres of the economic and non-economic superstructure of the state (also the social welfare), the preparation of the involved subjects in the state of security and an effective way of managing and coordinating their activities during the crisis situation. To support the fulfilment of these requirements the EM system utilises the centralised unified information system of economic mobilisation - UIS EM – and its specific application programme EPSIS® JISHM (furthermore only EPSIS). The employees of the Crisis Management Department worked out the technical and operating requirements in the process of procuring this information system (based on the assignment of the Ministry of Economy of the Slovak Republic).

UNIFIED INFORMATION SYSTEM OF ECONOMIC MOBILISATION

The specific programme EPSIS is a supporting programme that is used for the purposes of Economic Mobilization and for the crisis management in the Slovak Republic subjects, pointing especially at the current state and use of the specific application program of the "The Unified Information System of Economic Mobilization" with the trade mark EPSIS® JISHM – in Slovak Republic and its development in respect to user's requirements and building of resilience in the whole system. Some of these subjects are also bodies of the crisis management system. It serves for making decisions of the crisis management bodies during preparation for solutions as well as during the solution of a crisis situation itself and reducing the impacts of a crisis situation. The information system is only one aid which is to make the crisis manager's operation during making decisions for coping with the crisis situation or solving the crisis situation easier. During the state of security, the programme EPSIS is used for planning the realisation of the EM measures and for collecting data about the necessary resources (about the staff, material, energy, transportation, etc.) as well as for collecting data about the subjects and objects that can be used for solving the crisis situation.

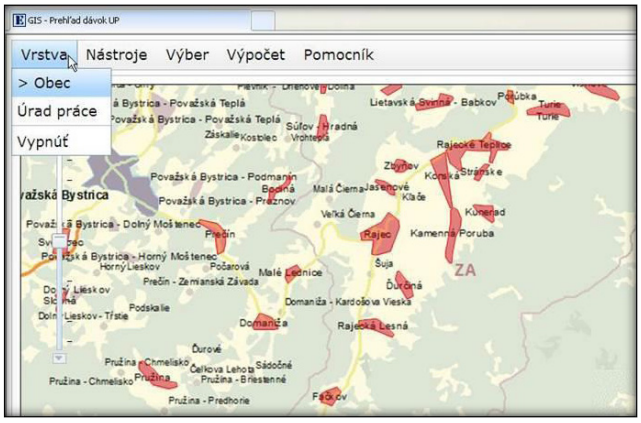


Figure 1 Map of assigned social benefits by the labour offices for the inhabitants of the affected villages in the framework of solving the crisis situation in EPSIS

Legend (to the Figure 1):
GIS – Prehľad dávok UP = GIS – Benefits overview of the Government Office of Labour, Social Affairs and Family of the Slovak Republic support, Obec = Village (Commune), Úrad práce = Government Office of Labour, Social Affairs and Family of the Slovak Republic, Vypnúť = Exit (the programme), Nástroje = Tools, Výber = Selection,
Výpočet = Calculation, Pomocník = Help.

The Ministry of Economy of the Slovak Republic creates a basic model composed of general modules which can be extended by specific modules based on the requirements of other EM subjects. These models can be used as a supporting materials and background for exercises within the constructive simulation in crisis management. They submit their requirements regarding to realising specific EM measures and it requests to include specific functions enabling to realise the given measures to the system. E.g. the Ministry of Labour, Social Affairs and Family of the Slovak Republic and cooperating EM subjects need such modules, functions and depictions (see the figure 1) which would enable ensuring the social services during the crisis situation effectively. On the other hand, the development of the programme EPSIS is limited by time and financial possibilities of individual administrators of the budget chapters.
In addition to ensuring the preparation of concerned subjects, it is also important to prepare controlling units, i.e. members of the crisis staff, to ensure an effective way of managing and coordinating all activities at the time of the crisis. These components also need to create favorable conditions in the field of preparation for resolving the crisis. An appropriate tool that can be implemented to support the training of crisis managers and community crisis staff members themselves is simulation. There are examples of trainings which were realized by so called „live simulation". One of the biggest trainings that took place in Slovakia was the training in 2012 called Havran. 4,800 people in total from various areas, such as crisis management, rescue services, firefighters, police, or even the CSSA, took part in this training. However, such large-scale trainings can be performed 1, up to 2 times a year. Our goal is to create a platform for practicing trainings that can be done in smaller groups and more often. One of the main task of our team is to develop models and scenarios that can be used as supporting materials and background for trainings within the constructive simulation in crisis management. We use this approach in our specialized Laboratory of Modelling

and Simulation of Crisis Phenomena to Improve Efficiency of Decision Making at the University Science Park of the University of Žilina.

CONCLUSION

The need to support making decisions by information technologies is nowadays an inseparable part of the decision-making process. EPSIS does not only replace the common information systems utilised by the individual EM subjects for solving their specific tasks and everyday administration activities but it enables coordinating activities and management for solving the crisis situations from the point of view of the central body (or a supervisory body – e.g. Centre of Labour, Social Affairs and Family) and executive subjects can (in real time) require completing the personal and material resources for carrying out the EM measures. In the framework of the social welfare they are e.g. completing the technical means for the CCSW operation, completing the human capacity necessary for settling the citizens' requirements, etc.

The University of Žilina's place in the framework of the Slovak citizens' social welfare is especially its active participation in the crisis management exercises where the university representative, namely the representatives of the Faculty of Security Engineering, play the role of referees. They supervise the exercises from the professional viewpoint and provide the participants with feedback in the form of recommendations for more effective management of the social welfare of the affected citizens and according to the exercise orientation also other areas. The subsequent collaboration can help implement the ideas to the methodological instructions for establishing, operating and managing the CCSWs. Also the programme EPSIS is similarly modified and is completed by suitable modules, functionalities and elements. Their goal is to solve the social situation and on the other hand to achieve the required performance of the affected citizens, i.e. to help them in their social situation as quickly as possible and in a suitable form. For details and references see (Zanicka Holla, K. and Moricova, V., 2011), (Holla, K., 2007), (Lovecek, T. et al., 2016), (Dvorak Z. et al., 2017) and (Vichova K. et al., 2017).

ACKNOWLEDGEMENT

"This article was created as a one of research project outcomes VEGA 1/0749/16 Risk Assessment and Treatment of Industrial Processes in Relation with Integrated Security and Safety within lower tier establishments."

"This work was supported by the research project VI20152019049 "RESILIENCE 2015: Dynamic Resilience Evaluation of Interrelated Critical Infrastructure Subsystems", supported by the Ministry of the Interior of the Czech Republic in the years 2015-2019."

"This work was jointly supported by the Slovak Research and Development Agency under the contract No. SK-CN-2017-0023."

"This work was supported by the Internal Grant Scheme of Faculty of Security Engineering, University of Zilina from the grant No. IGP201805. "

The views expressed, however, are solely those of the authors and not necessarily those of the institutions with which they are affiliated or of their funding sources. The authors are solely responsible for any errors or omissions.

REFERENCES

Act No. 179/2011 Coll. *on Economic Mobilization* and amended in Act No. 387/2002 Coll. *on Governance in Crisis Situation Outside Time of War and Warfare*, as amended.

Decree of the Ministry of Economy of the Slovak Republic No. 385/2011 Coll., *which implements certain provisions of Act No. 179/2011 Coll. on Economic Mobilization*, as amended in Act No. 387/2002 Coll. *on Governance in Crisis Situation Outside Time of War and Warfare*, as amended.

Dvorak Z., Sventekova E., Rehak D., Cekerevac Z., 2017: *Assessment of Critical Infrastructure Elements in Transport*. In 10th International Scientific Conference Transbaltica 2017: Transportation Science and Technology. Procedia Engineering, 2017, Vol. 187, pp. 548-555. ISSN 1877-7058. DOI: 10.1016/j.proeng.2017.04.413

Holla, K., 2007: *Dealing with Key Terms in Risk Analysis and Phenomenon of Uncertainty in this Process*, Communications - Scientific Letters of the University of Zilina, vol. 9, No. 4, pp. 59-61, ISSN 1335-4205.

Hudák J., Vagaš V., Ambrová M., 2013: *Centre of Social Warfare as Effective Tool for Social Warfare in Crisis Situations*. In: Krízová manažment, Vol. 12, No. 2, pp. 75-79, ISSN 1336-0019.

Lovecek, T. et al., 2016: *Level of protection of critical infrastructure in the Slovak Republic*. In: International Conference on Engineering Science and Production Management (ESPM), 2016.

Methodological Guideline of the Ministry of Labour, Social Affairs and Family of the Slovak Republic No. 16053/2014 Coll. of the activities of the individual entities involved in the activities the Centre of Concentrated Social Welfare. 2014.

Šimák L., 2016: *Crisis Management in Public Administration*. Žilina, EDIS – UNIZA, ISBN 978-80-554-1165-1.

Ristvej J., et al. 2016: *Simulation Technologies in Risk Prevention within Crisis Management* In: ESM´2016 = Modelling and simulation 2016: the European simulation and modelling conference 2016 : October 26-28, 2016 Las Palmas, Gran Canaria, Spain. - Ostend: EUROSIS-ETI, p. 4, 2016. - ISBN 978-90-77381-95-3.

Ristvej J., et al. 2013: *Modelling, simulation and information systems as a tool to support decision-making process in crisis management*, In: ESM 2013 = Modelling and Simulation 2013 : the European Simulation and Modelling Conference 2013 : October 23-25, 2013 Lancaster, United Kingdom. - [S.l.: EUROSIS-ETI], 2013. P. 71-76, - ISBN 978-90-77381-79-3.

Vichova K., Hromada M., Rehak D., 2017: *The Use of Crisis Management Information Systems in Rescue Operations of Fire and rescue system in the Czech Republic*. In 12th international scientific conference of young scientists, Ph.D. students and their tutors (TRANSCOM 2017), Procedia Engineering, 2017, Vol. 192, pp. 947-952. ISSN 1877-7058. DOI: 10.1016/j.proeng.2017.06.163

Zanicka Holla, K., Moricova, V., 2011: *Human Factor Position in Rise and Demonstration of Accidents*, Communications - Scientific Letters of the University of Zilina, vol. 13, No. 2, pp. 49-52, ISSN 1335-4205.

BIOGRAPHY

Jozef RISTVEJgained his Ph.D. in 2007 at the University of Zilina, Slovakia – the EU, in Crisis Management. Since that time he received postdoctoral position as the Assistant Professor and in 2014 after successful habilitation he has received position as Associate Professor at the Department of Crisis Management, Faculty of Security Engineering, University of Zilina. Since October 2014 is at the position of Vice-Rector for International Relations and Marketing at the university.

In 2007 he participated in ISCRAM Crisis Management Summer School for Ph.D. students. In 2010 received the visiting scholar grant of Ruth Crawford Mitchell to the Center for Disaster Management at the Graduate School of Public and International Affairs, University of Pittsburgh.

In his work he is focusing on supporting the information systems in the area of the decision making processes in crisis management.

For the year of 2012 was selected by the European Commission as Erasmus Staff Ambassador - 25th anniversary of the Erasmus Programme. He is a member of editorial board of the Infotechnology journal in Lithuania, in 2012 was member of ISCRAM 2012 Conference organizing committee. He is author of several papers in Europe and abroad. He is a member of several scientific project teams and as well as of the ISCRAM Association in Brussels (Information Systems for Crisis Response and Management) and the Society for Science and the Public, in Washington D.C.

e-mail: Jozef.Ristvej@fbi.uniza.sk

AUTHOR LISTING

AUTHOR LISTING

Aghezzaf E.-H. **95**
Allison C. **177**
Arend M.G. **88**

Bachir A. **164/187**
Baetens J. **123**
Ben Aissa Y. **187**
Ben Smida M. **164**
Bhatnagar S. **28**
Bonello A. **213**
Boode W. **157**
Boyrie F. **103**
Bruneel H. **123**
Büchter H. **147**
Burgert F.L. **88**

Cabrera J.J.H. **171**
Cappello G. **112**
Cardoso J. **112**
Castro R. **44**
Claeys D. **95/123**
Clausen U. **139**
Conley W. **5/11**

Darendeliler A. **95**
De Vos B. **131**
Desai M. **28**
Deschamps H. **112**
Dupleac D. **177**

Foguelman D. **44**

Galuszka A. **17**
Gómez J. É. **171**
Grossschmidt G. **63/69**
Grzejszczak T. **17/49**

Harf M. **63/69**
Hellander A. **22**
Hutchinson K. **157**

Ivaschenko A. **192**

Jánošíková M. **219**
Jeong D. **82**

Kalantar P. **88**
Karanjkar N. **28**
Kargapolova N. A. **199**
Kaufmann D. **77**
Kehoe J. **54**
Khalgui M. **164/187**
Khatab A. **95**
Kocerka J. **17**
Korchivoy S. **192**
Krause M. **77**

Latos B.A. **88**
Limère V. **106**

Mayer I. **157**
Moller R. **49**
Mosbahi O. **187**
Mütze-Niewöhner S. **88**

Naidji I. **164**
Naumann S. **147**
Nistor-Vlad R.-M. **177**
Nitsch V. **88**

Olczyk A. **17**
Ondrejka R. **219**
Osterloh T. **77**

Pech-Gourg N. **103**
Perez-Ferragut M. **177**
Prisecaru I. **177**
Przybysz P.M. **88**

Raa B. **131**
Rao D. M. **205**
Ristvej J. **219**
Rossmann J. **77**
Roth R. **49**

Santos C. **157**
Schmid N. A. **106**
Schröder K.-U. **77**
Seo Y. **82**
Serieye T. **103**
Šimák L. **219**
Singh P. **22**
Siron P. **112**
Smieja J. **17**
Steyaert B. **123**
Studziński J. **39**
Suda D. **213**
Szeląg B. **39**

Titko M. **219**

AUTHOR LISTING

Vera S.M. **171**

Warmelink H. **157**
Wijnant H. **106**

Zhang D. **139**