5th Student Conference on Operational Research

SCOR 2016, April 8–10, 2016, Nottingham, UK

Edited by
Bradley Hardy
Abroon Qazi
Stefan Ravizza

Editors

Bradley Hardy	Abroon Qazi	Stefan Ravizza
Cardiff University	Strathclyde Business School	IBM Global Business Services
HardyB@cardiff.ac.uk	abroon.qazi@strath.ac.uk	stefan.ravizza@ch.ibm.com

ACM Classification 1998
B.8 Performance and Reliability, D.2.2 Design Tools and Techniques, F.2.2 Nonnumerical Algorithms and Problems, G.1.6 Optimization, G.1.10 Applications, G.2.3 Applications, I.2.6 Learning, I.2.8 Problem Solving, Control Methods, and Search, I.6 Simulation and Modelling, I.6.4 Model Validation and Analysis

ISBN 978-3-95977-004-0

Published online and open access by
Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at http://www.dagstuhl.de/dagpub/978-3-95977-004-0.

Publication date
August, 2016

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Digital Object Identifier: 10.4230/OASIcs.SCOR.2016.0

OASIcs – OpenAccess Series in Informatics

OASIcs aims at a suitable publication venue to publish peer-reviewed collections of papers emerging from a scientific event. OASIcs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

ISSN 2190-6807

www.dagstuhl.de/oasics

Contents

5th Student Conference on Operational Research (SCOR'16).
Editors: Bradley Hardy, Abroon Qazi, and Stefan Ravizza
Open Access Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Preface

We are delighted to present the proceeding for the 5th Student Conference on Operational Research (SCOR 2016). The aim of SCOR is for students who are studying Operational Research, Management Science or a related field to share their work, practice their presentation skills, receive constructive feedback, and network in a relaxed and friendly environment.

The conference took place from Friday the 8th to Sunday the 10th of April 2016 at the University of Nottingham's Jubilee campus where we welcomed delegates from the UK, Europe and even further afield. Of those who attended, 43 presented their work across several streams which included: Data Envelopment Analysis, Graphs and Networks, Healthcare, Mathematical Programming, Metaheuristics, Multiple Criteria Decision Analysis, Operations Management, Optimisation, Safety and Reliability Analysis, Simulation, Stochastic Modelling, Transport and Logistics, and Vehicle Routing. This year's prize and runner-up award for Best Presentation (sponsored by Nationwide) were presented to Carlos Lamas-Fernandez and Jussi Rasku, respectively.

Along with our delegates' talks we were also happy to have plenary talks from four excellent invited speakers: Louise Allison (The OR Society), Ralf Keuthen (EY), Timo P. Kunz (Net-a-Porter.com) and Richard Wood (Prinicipality). These plenary talks covered a range of topics from outlining the benefits of OR Society membership, a series of case studies highlighting how OR is used in the consulting industry, the desirable skills OR students have for careers outside of academia, and how OR techniques can be used for contingency planning against bio-terrorism. Returning from 2014, we welcomed back Piero Vitelli (Island41) to round off our first day with a shortened version of his "Potential Energy" workshop on presentation skills.

On behalf of the organising committee, we'd like to thank all of our sponsors and partners. This year we were blessed with sponsorship from the OR Society, EY, Nationwide, Prospect Recruitment, and Gower Optimal Algorithms Ltd. (G.O.A.L.), along with being partnered with the National Taught Course Centre in Operational Research (NATCOR) and the University of Nottingham. The conference never would have taken place without the support and encouragement of these institutions.

Finally, we'd like to thank the authors who submitted a paper for this year's proceedings. The review process was based on the presentation, quality and originality of the research and there were at least two referees assigned to each paper.

Thank you to everyone who was involved in SCOR 2016, this truly is a conference made special by those who attend it.

Bradley Hardy
Abroon Qazi
Stefan Ravizza

5th Student Conference on Operational Research (SCOR'16).
Editors: Bradley Hardy, Abroon Qazi, and Stefan Ravizza
Open Access Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Organisation

Conference Committee

- Bradley Hardy, Cardiff University, UK (Conference Chair)
- Kingshuk Jubaer Islam, University of Nottingham, UK (Local Chair)
- Pedro Crespo Del Granado, ETH Zurich, Switzerland (Chair of Steering Committee)
- Tim Gruchmann, Witten/Herdecke University, Germany
- Samuel Luen-English, Cardiff University, UK
- Wasin Padungwech, Cardiff University, UK
- Abroon Qazi, Strathclyde Business School, UK
- Andrea Taverna, Università degli Studi di Milano, Italy
- Penny Holborn, University of South Wales, UK
- Stefan Ravizza, IBM Global Business Services, Switzerland
- Martim Moniz, Université Libre de Bruxelles, Belgium

Sponsors

- The OR Society
- EY
- Nationwide Building Society
- Gower Optimal Algorithms Ltd. (G.O.A.L.)
- Prospect Recruitment

Partners

- A National Taught Course Center in Operational Research (NATCOR)
- The University of Nottingham

5th Student Conference on Operational Research (SCOR'16).
Editors: Bradley Hardy, Abroon Qazi, and Stefan Ravizza

Open Access Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

List of Authors

Antuela Tako
School of Business and Economics,
Loughborough University
Leicestershire, LE11 3TU, UK
a.takou@lboro.ac.uk

Christina Latsou
Department of Aeronautical and Automotive
Engineering, Loughborough University
Leicestershire, LE11 3TU, UK
c.latsou@lboro.ac.uk

Diana Roman
Department of Mathematics, College of
Engineering Design and Physical Science,
Brunel University
London, UB83PH, UK
Diana.Roan@brunel.ac.uk

Fabian Mett
Department of Mathematical Optimization,
Zuse Institute Berlin (ZIB)
Takustraße 7, 14195 Berlin, Germany
mett@zib.de

Fabio D'Andreagiovanni
Department of Mathematical Optimization,
Zuse Institute Berlin (ZIB)
Takustraße 7, 14195 Berlin, Germany
d.andreagiovanni@zib.de

Hanjing Zhang
School of Business and Economics,
Loughborough University
Leicestershire, LE11 3TU, UK
h.zhang@lboro.ac.uk

Jiyin Liu
School of Business and Economics,
Loughborough University
Leicestershire, LE11 3TU, UK
j.y.liu@lboro.ac.uk

Jonad Pulaj
Department of Mathematical Optimization,
Zuse Institute Berlin (ZIB)
Takustraße 7, 14195 Berlin, Germany
pulaj@zib.de

Jussi Rasku
Department of Mathematical Information
Technology, University of Jyväskylä
P.O. Box 35, FI-40014, Finland
jussi.rasku@jyu.fi

Lisa M. Jackson
Department of Aeronautical and Automotive
Engineering, Loughborough University
Leicestershire, LE11 3TU, UK
l.m.jackson@lboro.ac.uk

Luigi De Giovanni
Dipartimento di Matematica, Università
degli Studi di Padova
via Trieste 63, 35121 Padova, Italy
luigi@math.unipd.it

Manon Raap
Universität der Bundeswehr München,
Fakultät für Informatik
Werner-Heisenberg-Weg 39, 85577
Neubiberg, Germany
manon.raap@unibw.de

Martin Zsifkovits
Universität der Bundeswehr München,
Fakultät für Informatik
Werner-Heisenberg-Weg 39, 85577
Neubiberg, Germany
martin.zsifkovits@unibw.de

Martine Labbé
GOM, Université Libre de Bruxelles
Bd du Triomphe CP210/01, 1050 Bruxelles,
Belgium
mlabbe@ulb.ac.be

Maximilian Moll
Universität der Bundeswehr München,
Fakultät für Informatik
Werner-Heisenberg-Weg 39, 85577
Neubiberg, Germany
maximilian.moll@unibw.de

5th Student Conference on Operational Research (SCOR'16).
Editors: Bradley Hardy, Abroon Qazi, and Stefan Ravizza
Open Access Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Mohd A. Maasar
Department of Mathematics, College of Engineering Design and Physical Science, Brunel University
London, UB83PH, UK
Mohd.Maasar@brunel.ac.uk

Naoum Tsioptsias
School of Business and Economics, Loughborough University
Leicestershire, LE11 3TU, UK
n.tsioptsias@lboro.ac.uk

Nysret Musliu
Institute of Information Systems, Vienna University of Technology
A-1040 Vienna, Austria
musliu@dbai.tuwien.ac.at

Paresh Date
Department of Mathematics, College of Engineering Design and Physical Science, Brunel University
London, UB83PH, UK
Paresh.Date@brunel.ac.uk

Rundong Yan
Department of Aeronautical and Automotive Engineering, Loughborough University
Leicestershire, LE11 3TU, UK
r.yan@lboro.ac.uk

Sarah J. Dunnett
Department of Aeronautical and Automotive Engineering, Loughborough University
Leicestershire, LE11 3TU, UK
s.j.dunnett@lboro.ac.uk

Stefan Pickl
Universität der Bundeswehr München, Fakultät für Informatik
Werner-Heisenberg-Weg 39, 85577
Neubiberg, Germany
stefan.pickl@unibw.de

Stewart Robinson
School of Business and Economics, Loughborough University
Leicestershire, LE11 3TU, UK
s.l.robinson@lboro.ac.uk

Tommi Kärkkäinen
Department of Mathematical Information Technology, University of Jyväskylä
P.O. Box 35, FI-40014, Finland
tommi.karkkainen@jyu.fi

Veronica Dal Sasso
Dipartimento di Matematica, Università degli Studi di Padova
via Trieste 63, 35121 Padova, Italy
dalsasso@math.unipd.it

Utilizing Dual Information for Moving Target Search Trajectory Optimization*

Manon Raap[1], Maximilian Moll[2], Martin Zsifkovits[3], and Stefan Pickl[4]

1 **Universität der Bundeswehr München, Fakultät für Informatik, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany**
manon.raap@unibw.de

2 **Universität der Bundeswehr München, Fakultät für Informatik, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany**
maximilian.moll@unibw.de

3 **Universität der Bundeswehr München, Fakultät für Informatik, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany**
martin.zsifkovits@unibw.de

4 **Universität der Bundeswehr München, Fakultät für Informatik, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany**
stefan.pickl@unibw.de

Abstract

Various recent events have shown the enormous importance of maritime search-and-rescue missions. By reducing the time to find floating victims at sea, the number of casualties can be reduced. A major improvement can be achieved by employing autonomous aerial systems for autonomous search missions, allowed by the recent rise in technological development. In this context, the need for efficient search trajectory planning methods arises. The objective is to maximize the probability of detecting the target at a certain time k, which depends on the estimation of the position of the target. For stationary target search, this is a function of the observation at time k. When considering the target movement, this is a function of *all* previous observations up until time k. This is the main difficulty arising in solving moving target search problems when the duration of the search mission increases. We present an intermediate result for the single searcher single target case towards an efficient algorithm for longer missions with multiple aerial vehicles. Our primary aim in the development of this algorithm is to disconnect the networks of the target and platform, which we have achieved by applying Benders decomposition. Consequently, we solve two much smaller problems sequentially in iterations. Between the problems, primal and dual information is exchanged. To the best of our knowledge, this is the first approach utilizing dual information within the category of moving target search problems. We show the applicability in computational experiments and provide an analysis of the results. Furthermore, we propose well-founded improvements for further research towards solving real-life instances with multiple searchers.

1998 ACM Subject Classification G.2.3 Applications

Keywords and phrases Search Theory, UAV, Vehicle Routing, Benders Decomposition

Digital Object Identifier 10.4230/OASIcs.SCOR.2016.1

* This work was supported by Airbus Defence & Space.

5th Student Conference on Operational Research (SCOR'16).
Editors: Bradley Hardy, Abroon Qazi, and Stefan Ravizza; Article No. 1; pp. 1:1–1:10

Open Access Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

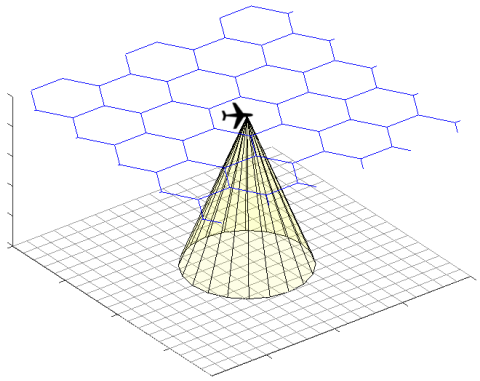

Figure 1 The heterogeneous state spaces for the target (square grid) and the platform (hexagonal grid).

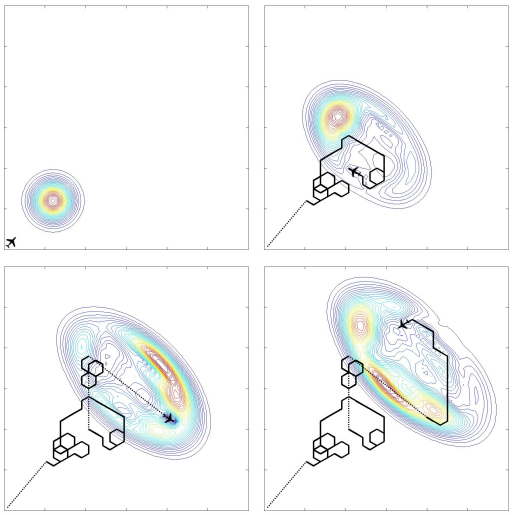

Figure 2 Snapshots of a simulation of one fixed-wing platform searching for one Markovian target. The contour plot indicates the updated probability distribution of the target location.

1 Introduction

In the spring of 2015, over 1,000 lifes were lost at sea in a single month after refugee ships sunk in the Mediterranean. These tragedies led to renewed efforts by governmental and non-governmental organizations that significantly decreased the numbers of deaths [27]. Search missions are often supported by manned aerial vehicles where individual pilots are expected to plan their optimal search trajectory by hand, which is tremendously complex; it is proven to be an $\mathcal{NP}$-complete optimization problem [26] for a single platform searching for a single *stationary* target. Planning for *moving* target search is considerably more complex in general. Executing such a complex task in a stressful situation is susceptible to result in a sub-optimal search trajectory. We therefore aim to automatize this task, with the outlook towards autonomous search missions by unmanned aerial vehicles (UAVs). We refer to an aerial sensor platform (e.g. UAV) by *platform* for short in the remainder of this article.

The contribution of this article consists of a Benders decomposition algorithm to solve a problem of Markovian target search trajectory optimization (see Figure 2). We decompose an existing mixed integer linear program (MILP) [19] that accounts for a heterogeneous state space for the target and platform. This concept is shown in Figure 1 and holds two benefits; a target specific grid allows for a more accurate estimation of the target position, whereas a platform specific grid allows for modeling more natural flight motion. By applying Benders decomposition, we are able to disconnect both networks and iteratively solve two much smaller problems until the desired optimality tolerance has been reached. A diagrammatic visualization of this decomposition is shown in Figure 3. To the best of our knowledge, this is the first method utilizing the dual information for solving a moving target search problem.

The remainder of this article is structured as follows: Section 2 provides an overview of the relevant related literature. In Section 3, the search trajectory problem is stated, followed by our methodology for solving this problem in Section 4. Computational experiments in Section 5 show the applicability of the proposed method. Finally, the conclusions are presented in Section 6.

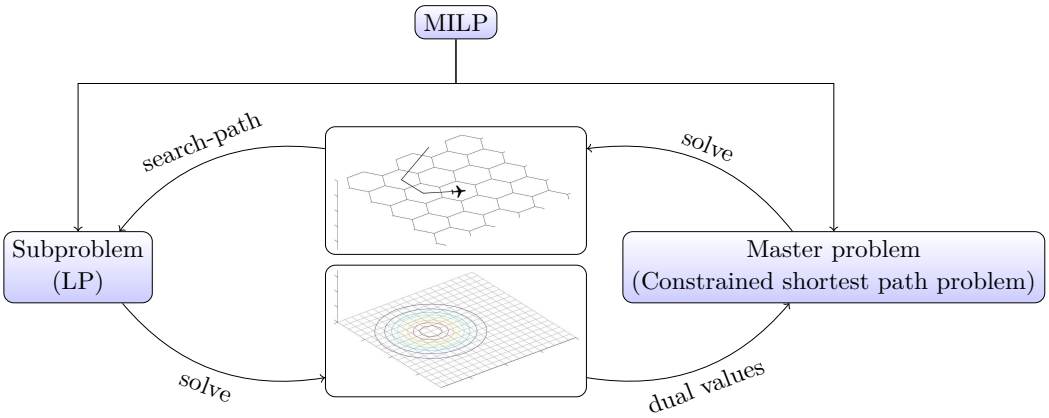

Figure 3 The MILP decomposes into a subproblem and a master problem. The master problem is a constrained shortest path problem on the platform network and the subproblem is a linear problem to calculate the probability of detection and containment on the target network, given a fixed search-path provided by the master. In turn, the subproblem provides the dual values to the master problem. As a result, the networks are disconnected and we solve two much smaller problems sequentially in iterations.

2 Related Literature

Research initiated by Koopman [13] has led to the interesting field of *Search Theory* [24, 25]. To apply search theory to real life search missions, efficient planning methods are necessary. Extensive research has led to several search strategy planning approaches, which can mainly be classified to two problem types. For the path-constrained search effort allocation problem [23], the search area is typically divided into subareas to which search effort is allocated over time. The search effort is expressed in number of platforms of a certain type and duration. For autonomous search, the kinematical properties of the sensor platform must be taken into account. The resulting problem is referred to by the search trajectory optimization problem. A depth-first branch and bound approach was presented in [23], in which lower bound approximations on the probability of non-detection are obtained by relaxing the searcher's path constraints. Most following approaches are of the branch and bound type [6, 7, 15, 17, 21], where the aim is to find a tightest true lower bound with low computational costs. Computational experiments of several branch and bound procedures are summarized in [28]. Despite vast progression over the years, computation remains intractable for larger instances. This has led to the development of heuristic approaches such as a receding horizon approach [5], cross entropy optimization [14] and constraint programming [18].

This article describes a Benders decomposition [11, 16] algorithm for a trajectory optimization problem for aerial vehicles. Also various other vehicle routing problems have been tackled by a Benders decomposition algorithm, i.e. [1, 4]. Algorithms of this type have also been developed for *combined* vehicle routing problems with allocation [2, 8], assignment [3, 9], and scheduling [22].

3 The Moving Target Search Trajectory Problem

We consider the search for a moving target in discrete time on a finite set of cells $\mathcal{C}$. This is a well known problem in literature and can be formally stated as follows. The target occupies one unknown cell $C_k \in \mathcal{C}$ at time step $k \in \mathcal{K}$. For the duration of the search mission,

a probability map $\boldsymbol{pc}$ is maintained, where the probability of containment $pc_k(c) \in [0,1]$ represents the probability of the target occupying cell c at time k, without having been detected prior to time k. Although the initial position of the target is unknown, it is characterized by a known prior probability distribution $\boldsymbol{pc}_1$. The *target* path is modeled by a stochastic process $(C_1, ..., C_K)$, which we assume to be Markovian. Under this assumption, the probability map evolves due to the target motion according to

$$pc_{k+1}(c) = \sum_{c' \in \mathcal{C}} d(c', c)\, pc_k(c'), \tag{1}$$

where the transition function $d(c', c) \in [0,1]$ represents the probability that the target moves from cell c' to cell c and is assumed to be known for each pair $(c', c) \in \mathcal{C} \times \mathcal{C}$.

The *platform* is modeled to move over a finite grid of hexagonal shape. This grid is represented by a network $G = (\mathcal{V}, A)$ with the set of nodes $\mathcal{V}$ and binary adjacency matrix A. Here, an entry $a_{v,v'}$ of A is 1 if node $v \in \mathcal{V}$ is adjacent to node $v' \in \mathcal{V}$ and 0 otherwise. We assume that the considered aerial sensor platform has a sensor equipped to make observations. The glimpse probability $pg_k(v_k, c) \in [0,1]$ represents the probability of target detection, given target occupancy within cell c and platform position $v_k \in \mathcal{V}$ at time k. When observations are made, the probability map evolves according to the motion model as in Equation (1) and, in addition, evolves according to the glimpse probability due to the observations made when visiting node v_k. Therefore, Equation (1) is extended to account for observation results as follows:

$$pc_{k+1}(c) = \sum_{c' \in \mathcal{C}} d(c', c)\, pc_k(c')\, (1 - pg_k(v_k, c'))\,. \tag{2}$$

The *objective* is to determine a sequence of nodes $\boldsymbol{v} = (v_1, \ldots, v_K)$ maximizing the cumulative probability of detection over time period $\mathcal{K}$, i.e.

$$\max \sum_{k=1}^{K} \sum_{c \in \mathcal{C}} pd_k(v_k, c), \tag{3}$$

where $pd_k(v_k, c)$ is the probability of detecting the target at time k from platform position v_k in cell c and is calculated by

$$pd_k(v_k, c) = pc_k(c)\, pg_k(v_k, c), \tag{4}$$

where the probability of containment $pc_k(c)$ is calculated through Equation (2).

3.1 Mixed Integer Linear Programming Formulation

The model for moving target search trajectory optimization under kinematical constraints can be stated as the following MILP. Here, let $\mathbb{B} = \{0, 1\}$. The decision variable $z_v^k \in \mathbb{B}$ is 1 if the platform is at node v at time k and 0 otherwise. The auxiliary decision variable $pd_c^k \geq 0$ represents the probability of detection in cell c at time k. The auxiliary decision variable $pc_c^k \geq 0$ represents the probability of containment in cell c at time k. We use *italic* font for all decision variables and normal font for input variables.

$$\text{Maximize} \sum_{k=1}^{K} \sum_{c \in \mathcal{C}} pd_c^k \tag{5}$$

subject to

$$pd_c^k - \mathrm{pg}_{v,c}^k pc_c^k \leq 1 - z_v^k \qquad \forall k \in \mathcal{K}, \forall v \in \mathcal{V}, \forall c \in \mathcal{C} \tag{6}$$

$$pd_c^k - \sum_{v\in\mathcal{V}} \mathrm{pg}_{v,c}^k z_v^k \leq 0 \qquad \forall k \in \mathcal{K}, \forall c \in \mathcal{C} \tag{7}$$

$$pc_c^k - \sum_{c'\in\mathcal{C}} \mathrm{d}_{c',c} pc_{c'}^{k-1} + \sum_{c'\in\mathcal{C}} \mathrm{d}_{c',c} pd_{c'}^{k-1} = 0 \qquad \forall k \in \{2,\dots,K\}, \forall c \in \mathcal{C} \tag{8}$$

$$\boldsymbol{z} \in \boldsymbol{Z} \tag{9}$$

$$pc_c^1 = \mathrm{pc}_c^1 \qquad \forall c \in \mathcal{C} \tag{10}$$

$$z_v^k \in \mathbb{B} \qquad \forall k \in \mathcal{K}, \forall v \in \mathcal{V} \tag{11}$$

$$pd_c^k, pc_c^k \geq 0 \qquad \forall k \in \mathcal{K}, \forall c \in \mathcal{C} \tag{12}$$

The objective function (5) maximizes the cumulative probability of detection. The sets of constraints in (6) and (7) ensure calculation of pd_c^k according to the formula (4). The set of constraints in (8) ensure calculation of pc_c^k according to the formula (2) and the set of constraints in (10) ensure that the initial values of pc_c^1 are set according to the known prior probability distribution $\boldsymbol{pc}_1$. Let $\boldsymbol{Z}$ be the set of binary vectors for the z_v^k variables that yield feasible trajectories on the network G. We use this abstract notation, so that feasibility of a trajectory can be defined as desired. The set of constraints (9) ensures the binary vector $\boldsymbol{z}$ to be a feasible trajectory. The total number of decision variables, including the auxiliary decision variables pd_c^k and pc_c^k, is of order $\mathcal{O}(K|\mathcal{V}||\mathcal{C}|)$. The number of possible feasible trajectories on network G is of order $\mathcal{O}(2^{K|\mathcal{V}|})$.

4 Solution Methodology

As the number of time steps or the number of nodes increases, the problem becomes more difficult to solve. In order to decrease the rate in which computation time increases, we decompose model (5)–(12) into a pair of problems that can be solved more easily by applying Benders decomposition [10]. The primal and dual subproblem, as well as the Benders master problem are stated in the following subsections.

4.1 Primal Subproblem

Recall $\boldsymbol{Z}$ to be the set feasible trajectories on the platform network G. When fixing any binary vector $\bar{\boldsymbol{z}} \in \boldsymbol{Z}$, the original problem reduces to the following *primal subproblem* in the pc_c^k and pd_c^k variables. This linear program contains no integer decision variables and is therefore very easy to solve.

$$\text{Maximize} \sum_{k=1}^{K} \sum_{c\in\mathcal{C}} pd_c^k \tag{13}$$

subject to

$$pd_c^k - \mathrm{pg}_{v,c}^k pc_c^k \leq 1 - \bar{z}_v^k \qquad \forall k \in \mathcal{K}, \forall v \in \mathcal{V}, \forall c \in \mathcal{C} \tag{14}$$

$$pd_c^k - \sum_{v\in\mathcal{V}} \mathrm{pg}_{v,c}^k \bar{z}_v^k \leq 0 \qquad \forall k \in \mathcal{K}, \forall c \in \mathcal{C} \tag{15}$$

$$pc_c^k - \sum_{c'\in\mathcal{C}} \mathrm{d}_{c',c} pc_{c'}^{k-1} + \sum_{c'\in\mathcal{C}} \mathrm{d}_{c',c} pd_{c'}^{k-1} = 0 \qquad \forall k \in \{2,...,K\}, \forall c \in \mathcal{C} \tag{16}$$

$$pc_c^1 = \mathrm{pc}_c^1 \qquad \forall c \in \mathcal{C} \tag{17}$$

$$pd_c^k, pc_c^k \geq 0 \qquad \forall k \in \mathcal{K}, \forall c \in \mathcal{C} \tag{18}$$

4.2 Dual Subproblem

Let $\boldsymbol{\pi} = (\pi^k_{v,c} | k \in \mathcal{K}, v \in \mathcal{V}, c \in \mathcal{C})$, $\boldsymbol{\rho} = (\rho^k_c | k \in \mathcal{K}, c \in \mathcal{C})$, $\boldsymbol{\sigma} = (\sigma^k_c | k \in \{2, ..., K\}, c \in \mathcal{C})$, and $\boldsymbol{\tau} = (\tau^1_c | c \in \mathcal{C})$ be the dual variables associated with constraints (14), (15), (16), and (17) respectively. The dual of the *primal subproblem* (13)-(18) is the following *dual subproblem*:

$$\text{Minimize} \sum_{k=1}^{K} \sum_{c \in \mathcal{C}} \sum_{v \in \mathcal{V}} \pi^k_{v,c} \left(1 - \bar{z}^k_v\right) + \sum_{k=1}^{K} \sum_{c \in \mathcal{C}} \rho^k_c \left(\sum_{v \in \mathcal{V}} \text{pg}^k_{v,c} \bar{z}^k_v \right) + \sum_{c \in \mathcal{C}} \tau^1_c \text{pc}^1_c \tag{19}$$

subject to

$$\sum_{v \in \mathcal{V}} \pi^k_{v,c} + \rho^k_c + \mathbb{1}_{\{k \leq K-1\}} \sum_{c' \in \mathcal{C}} \sigma^{k+1}_{c'} d_{c,c'} \geq 1 \qquad \forall k \in \mathcal{K}, \forall c \in \mathcal{C} \tag{20}$$

$$\sum_{v \in \mathcal{V}} \text{pg}^k_{v,c} \pi^k_{v,c} - \mathbb{1}_{\{k \neq 1\}} \sigma^k_c + \mathbb{1}_{\{k \leq K-1\}} \sum_{c' \in \mathcal{C}} \sigma^{k+1}_{c'} d_{c,c'} \tag{21}$$

$$-\mathbb{1}_{\{k=1\}} \tau^1_c \leq 0 \qquad \forall k \in \mathcal{K}, \forall c \in \mathcal{C} \tag{22}$$

$$\pi^k_{v,c}, \rho^k_c \geq 0 \qquad \forall k \in \mathcal{K}, \forall v \in \mathcal{V}, \forall c \in \mathcal{C} \tag{23}$$

$$\sigma^k_c, \tau^1_c \in \mathbb{R} \qquad \forall k \in \{2, \ldots, K\}, \forall c \in \mathcal{C} \tag{24}$$

Here, function $\mathbb{1}$ denotes the indicator function so that $\mathbb{1}_{\{x\}} = 1$ if x is true and 0 otherwise. Let $\boldsymbol{\Delta}$ be the polyhedron defined by constraints (20)-(23), and let P_Δ denote the set of extreme points of $\boldsymbol{\Delta}$. Polyhedron $\boldsymbol{\Delta}$ contains no rays because the subproblem is feasible for each $\bar{\boldsymbol{z}} \in \boldsymbol{Z}$.

4.3 Benders Master Problem

We introduce the free decision variable $y_0 \in \mathbb{R}$ and reformulate the original model (5)-(12) as the following *Benders master problem*:

$$\text{Maximize } y_0 \tag{25}$$

subject to

$$\sum_{k=1}^{K} \sum_{c \in \mathcal{C}} \sum_{v \in \mathcal{V}} \pi^k_{v,c} \left(1 - \bar{z}^k_v\right) + \sum_{k=1}^{K} \sum_{c \in \mathcal{C}} \rho^k_c \left(\sum_{v \in \mathcal{V}} \text{pg}^k_{v,c} \bar{z}^k_v \right) \tag{26}$$

$$+ \sum_{c \in \mathcal{C}} \tau^1_c \text{pc}^1_c \geq y_0 \qquad \forall (\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\tau}) \in P_\Delta \tag{27}$$

$$\boldsymbol{z} \in \boldsymbol{Z} \tag{28}$$

$$z^k_v \in \mathbb{B} \qquad \forall k \in \mathcal{K}, \forall v \in \mathcal{V} \tag{29}$$

$$y_0 \in \mathbb{R} \tag{30}$$

The set P_Δ is very large in general, resulting in many constraints in the Benders master problem. However, most of these constraints are not active in an optimal solution. We therefore do not have to include all these constraints, but generate a subset in an iterative algorithm. In each iteration, a relaxed Benders master problem is solved, which is obtained by replacing the set P_Δ with the subset P^t_Δ of extreme points available at iteration $t = 0, \ldots$. At the start, this subset is empty, i.e. $P^0_\Delta = \emptyset$. Let $(\boldsymbol{z}, y_0)$ be an optimal solution to this problem. Next, the primal subproblem (13)-(18) is solved with the values of $\boldsymbol{z}$ being fixed. This yields the dual variables $(\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\tau})$ associated with constraints (14), (15), and (17) respectively.

Because the primal subproblem is feasible for each $\boldsymbol{z}$, the values of the dual variables $(\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\tau})$ determine an extreme point of P_Δ and we add this point to P_Δ^{t+1}. As a consequence, one constraint is added to the relaxed Benders master problem. This process continues until its optimal value equals the optimal value of the primal subproblem.

5 Computational Experiment

In this section, we present computational experiments to show the applicability of our algorithm. First, we present the test instances of our search problem, followed by the results and a concise analysis thereof.

The experiments were performed on an Intel(R) CoreTM i7-4810MQ CPU processor with 2.80 GHz and a usable memory of 15.6 GB. The instances were generated with our simulation platform, that is written in Matlab. The original MILP and the Benders decomposition algorithm were coded in Java and solved using CPLEX with default parameters. We set the maximum number of iterations in the Benders decomposition algorithm to 3000 to avoid out-of-memory exceptions.

5.1 Description of the Instances

In the instances generated for our analysis we consider one target and one platform. The target state space $\mathcal{C}$ consists of 60 square cells, over which the initial target position is normally distributed in two dimensions with its mean in the center and with a standard deviation of 12. The target moves in north direction with 0.5 probability and in east direction with 0.5 probability as well, with a speed of one cell per time step. The platform state space $\mathcal{V}$ consists of 20 nodes and its start position is trivial, since we allow the platform to reach any node in the first time step. It then searches with the speed of one node per time step. The ratio of the lengths of a square cell and a hexagonal cell is $2:1$. For sensor characteristics, we used the typical glimpse probability function [7]:

$$pg_k(v_k, c) = 1 - \exp^{-\omega(v,c,k)}, \tag{31}$$

with $\omega(v, c, k) \geq 0$ being a measure of search effectiveness for cell c. The search effectiveness decreases with the Euclidean distance $||v_k - c||$ between cell c and the platform at node v_k, as follows:

$$\omega(v, c, k) = W\left(||v_k - c||\right)^{-1},$$

where W is some sensor quality indicator. We used $W = 0.7$ and generated four instances with these properties, for time periods $K = (4, 6, 8, 10)$.

5.2 Results and Analysis

We optimized the search trajectory on the four instances with the proposed Benders decomposition (BD) algorithm, and additionally with CPLEX on the original formulation for benchmarking. An optimal trajectory was found for all instances by both algorithms, although we noticed differences in computation time. In Table 1, we list for each instance the time period (K), the optimal value in cumulative probability of detection (PD), the number of iterations in the BD algorithm, time to solve for the BD algorithm in seconds, total and average time spend on solving the relaxed Benders master problem (RBMP) and on the primal subproblem (PSP), and finally the time needed for CPLEX to solve the original formulation (MILP).

Table 1 Test results.

K	Optimal value (PD)	Iterations (#)	BD (s)	RBMP Tot (s)	RBMP Avg (s)	PSP Tot (s)	PSP Avg (s)	MILP (s)
4	0.2786	352	65.43	62.13	0.18	2.22	0.01	0.77
6	0.3213	884	182.88	152.25	0.17	11.25	0.01	2.22
8	0.3368	1980	1738.19	1535.37	0.78	113.43	0.06	7.97
10	0.3439	3000	8740.82	8171.55	2.72	288.66	0.10	15.46

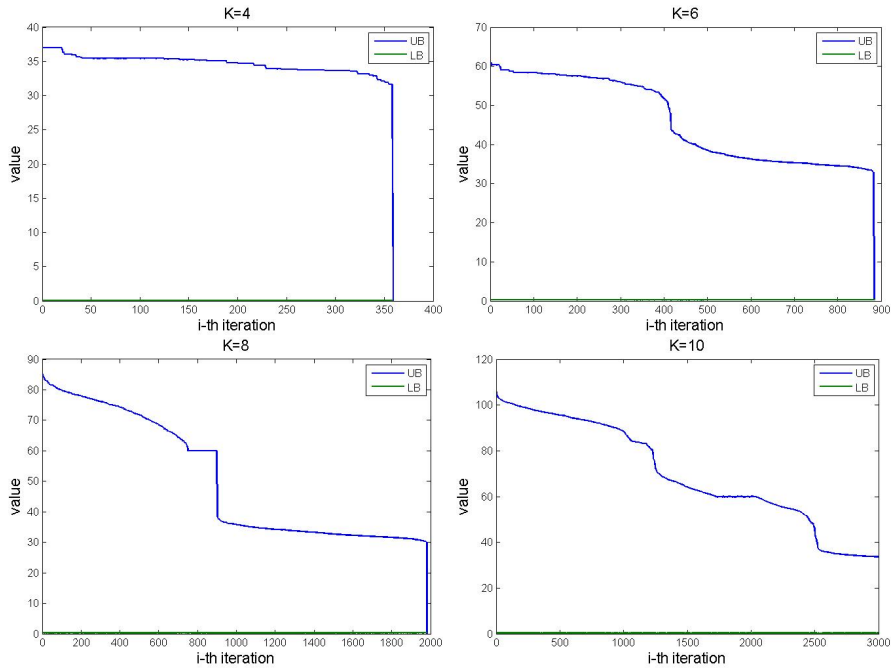

Figure 4 Convergence profiles of the instances $K = 4$ shown top-left to $K = 10$ shown bottom-right.

As the results in Table 1 show, the proposed BD algorithm is not faster than solving the original MILP. Even for the instance with $K = 10$, the cap of 3000 iterations was hit and the algorithm was terminated before it converged. The optimal solution was found, but not proved. The results however supply valuable information; the major portion of time is spend on solving the RBMP, whereas the PSP is very easily optimized. Also the convergence profiles (see Figure 4) show that many cuts are necessary to move out of local optima, of which we seem to have $(K/2) - 1$ in each instance. These observations lead us to the following considerations for further research.

5.3 Considerations for Further Research

There are several improvements to be considered, which show potential to lead to improved calculation times with respect to solving the original MILP:

- One improvement is to utilize local branching for Benders decomposition [20]. The nodes resulting from a branching step each represent a smaller subregion of the feasible region, so that cuts are found more easily. This method is applicable to any binary program that can be solved by BD and may reduce the computation time significantly.

- A major difficulty lies in solving the RBMP, which is a large 0-1 linear program with one continuous variable. Improvements can be made by decreasing the computation time on this problem, which is actually a well studied resource constrained longest path problem on a directed acyclic graph. It can be solved by specific and more efficient algorithms as described in [12], such as dynamic programming and Lagrangean relaxation. Also heuristics are applicable, because it is possible to generate new cuts from any feasible integer solution.
- A major strength of our formulation is that the PSP is an easy to solve LP. This may prove to be necessary when solving the multi-platform case using Benders decomposition. The multi-platform search trajectory problem can then be decomposed into multiple RBMPs together with a single PSP providing the dual information for each platform specific RBMP.

6 Conclusion

We considered the problem of optimizing a Markovian target search trajectory under kinematical constraints. For this problem, we proposed a novel method that utilizes dual information by applying Benders decomposition. The aim of this research was to develop a method that disconnects the networks of the target and platform and thereby solving much smaller problems in iterations. Results from computational experiments show this accomplishment. Possibilities for efficiently solving this problem for the multi-platform case arise, because the master problem decomposes into multiple smaller problems for each platform. This is of importance because the original formulation grows exponentially in the number of platforms. All known exact methods for solving the multi-platform case appear intractable for very small instances already and are therefore not applicable to real-life *cooperative* search missions. The network disconnection as accomplished in this work is a first step towards an exact method for solving the multi-platform case that grows only linearly in the number of platforms. Therefore, this is a significant step towards automated cooperative search missions by unmanned aerial vehicles.

Acknowledgements. This work was supported by Airbus Defence & Space, Germany.

References

1 Tolga Bektaş. Formulations and benders decomposition algorithms for multidepot salesmen problems with load balancing. *European Journal of Operational Research*, 216(1):83–93, 2012.

2 James H. Bookbinder and Kathleen E. Reece. Vehicle routing considerations in distribution system design. *European Journal of Operational Research*, 37(2):204–213, 1988.

3 Jean-François Cordeau, François Soumis, and Jacques Desrosiers. A benders decomposition approach for the locomotive and car assignment problem. *Transportation science*, 34(2):133–149, 2000.

4 Yvan Dumas, Jacques Desrosiers, Eric Gelinas, and Marius M. Solomon. An optimal algorithm for the traveling salesman problem with time windows. *Operations research*, 43(2):367–371, 1995.

5 James N. Eagle. *The approximate solution of a simple constrained search path moving target problem using moving horizon policies.* Monterey, California. Naval Postgraduate School, 1984.

6 James N. Eagle. The optimal search for a moving target when the search path is constrained. *Operations Research*, 32(5):1107–1115, 1984.

7 James N. Eagle and James R. Yee. An optimal branch-and-bound procedure for the constrained path, moving target search problem. *Operations Research*, 38(1):110–114, feb 1990.

8 Awi Federgruen and Paul Zipkin. A combined vehicle routing and inventory allocation problem. *Operations Research*, 32(5):1019–1037, 1984.

9 Marshall L. Fisher and Ramchandran Jaikumar. A generalized assignment heuristic for vehicle routing. *Networks*, 11(2):109–124, 1981.

10 Arthur M. Geoffrion. Generalized benders decomposition. *Journal of optimization theory and applications*, 10(4):237–260, 1972.

11 John N. Hooker and Greger Ottosson. Logic-based benders decomposition. *Mathematical Programming*, 96(1):33–60, 2003.

12 Stefan Irnich, Guy Desaulniers, et al. Shortest path problems with resource constraints. *Column generation*, 6730:33–65, 2005.

13 Bernard Osgood Koopman. *Search and screening: general principles with historical applications*. Pergamon Press, 1980.

14 P. Lanillos, E. Besada-Portas, G. Pajares, and J.J. Ruz. Minimum time search for lost targets using cross entropy optimization. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 602–609, oct 2012.

15 Haye Lau, Shoudong Huang, and Gamini Dissanayake. Discounted MEAN bound for the optimal searcher path problem with non-uniform travel times. *European Journal of Operational Research*, 190(2):383–397, 2008.

16 Richard Kipp Martin. *Large scale linear and integer optimization: a unified approach*. Springer Science & Business Media, 2012.

17 Gustavo H. A. Martins. A new branch and bound procedure for computing optimal search paths. Master's thesis, Naval Postgraduate School, Monterey, CA, 1993.

18 Michael Morin, Anika-Pascale Papillon, Irène Abi-Zeid, François Laviolette, and Claude-Guy Quimper. Constraint programming for path planning with uncertainty. In Michela Milano, editor, *Principles and Practice of Constraint Programming*, Lecture Notes in Computer Science, pages 988–1003. Springer Berlin Heidelberg, 2012.

19 M. Raap, S. Pickl, and M. Zsifkovits. Trajectory optimization under kinematical constraints for moving target search. *(working paper)*, 2016.

20 Walter Rei, Jean-François Cordeau, Michel Gendreau, and Patrick Soriano. Accelerating benders decomposition by local branching. *INFORMS Journal on Computing*, 21(2):333–345, 2009.

21 Hiroyuki Sato and Johannes O. Royset. Path optimization for the resource-constrained searcher. *Naval Research Logistics*, 57(5):422–440, 2010.

22 Thomas R. Sexton and Lawrence D. Bodin. Optimizing single vehicle many-to-many operations with desired delivery times: I. Scheduling. *Transportation Science*, 19(4):378–410, 1985.

23 T.J. Stewart. Search for a moving target when searcher motion is restricted. *Computers & Operations Research*, 6(3):129–140, 1979.

24 Lawrence D. Stone. *Theory of Optimal Search*. 1975.

25 Lawrence D. Stone, Johannes O. Royset, and Alan R. Washburn. *Optimal Search for Moving Targets*. Springer, 2016.

26 K. E. Trummel and J. R. Weisinger. The complexity of the optimal searcher path problem. *Operations Research*, 34(2):324–327, mar 1986.

27 UN High Commissioner for Refugees (UNHCR). The sea route to europe: The mediterranean passage in the age of refugees, 2015. Last accessed: 24 Feb 2016.

28 Alan R. Washburn. Branch and bound methods for a search problem. *Naval Research Logistics (NRL)*, 45(3):243–257, 1998.

Reliability Modelling of Automated Guided Vehicles by the Use of Failure Modes Effects and Criticality Analysis, and Fault Tree Analysis*

Rundong Yan[1], Sarah J. Dunnett[2], and Lisa M. Jackson[3]

1 **Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, Leics. LE11 3TU, U.K.**
r.yan@lboro.ac.uk
2 **Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, Leics. LE11 3TU, U.K.**
s.j.dunnett@lboro.ac.uk
3 **Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, Leics. LE11 3TU, U.K.**
l.m.jackson@lboro.ac.uk

Abstract

Automated Guided Vehicles (AGVs) are being increasingly used for intelligent transportation and distribution of materials in warehouses and auto-production lines. In this paper, a preliminary hazard analysis of an AGV's critical components is conducted by the approach of Failure Modes Effects and Criticality Analysis (FMECA). To implement this research, a particular AGV transport system is modelled as a phased mission. Then, Fault Tree Analysis (FTA) is adopted to model the causes of phase failure, enabling the probability of success in each phase and hence mission success to be determined. Through this research, a promising technical approach is established, which allows the identification of the critical AGV components and crucial mission phases of AGVs at the design stage.

1998 ACM Subject Classification B.8 Performance and Reliability

Keywords and phrases Reliability, Fault Tree Analysis, Automated Guided Vehicles, FMECA

Digital Object Identifier 10.4230/OASIcs.SCOR.2016.2

1 Introduction

The concept of an Automated Guided Vehicle (AGV), which travels along a predefined route without an on-board operator to perform prescribed tasks, was first introduced in 1955 [1]. Nowadays, AGVs are being increasingly used for intelligent transportation and distribution of materials in warehouses and/or manufacturing facilities attributed to their high efficiency, safety and low costs. As the AGV systems are getting larger and more complex, increasing the efficiency and lowering the operation cost of the AGV system has naturally become the first priorities, via identifying new flow-path layouts and developing advanced traffic management strategies (e.g. vehicle routing) [2]. For this reason, the previous research effort in the AGV area has mainly focused on route optimisation and traffic management of AGVs. For example, Giuseppe established an approach in 2013 to optimise the flow-path such that the average time for carrying out transportation tasks can be minimised and the utilisation

* This work was supported by EPSRC.

degree of AGVs can be maximised at the same time [3]; Wu and Zhou created a simulation model to avoid collisions, deadlock, blocking and minimise the route distance as well with a coloured resource-oriented Petri Net [4]. However, little effort has been made to investigate the safety and reliability issues of the AGV components/subassemblies and their probability of success in completing a prescribed mission. Although Fazlollahtabar recently created a model to maximise the total reliability of the AGVs and minimise the repair cost of AGV systems [5], they considered the AGV as a whole. Hence, fundamental questions, such as 'How could AGVs fail?' and 'What are the possibilities of their failure?', have not been answered.To answer these questions, Duran and Zalewski tried to identify the basic failure modes of the light detection and ranging (LIDAR) system and the camera-based computer vision system (CV) on AGVs in 2013 by the approach of Fault Tree Analysis (FTA) and Bayesian Belief Networks (BBN) [6]. In that work, human injury, property damage and vehicle damage were defined as the top events in the fault tree. However, the research did not cover all components and subassemblies in AGVs. A complete investigation of the safety and reliability issues of all AGV components and subassemblies is important not only to ensure the high reliability and availability of AGVs and their success of delivering prescribed tasks, but also to optimise their maintenance strategies. Research is conducted in this paper to identify the critical risks of all AGV components and the crucial mission phases in an AGV operation. Failure Modes Effects and Criticality Analysis (FMECA) and Fault Tree Analysis (FTA) will be adopted to achieve this. Hence, the contribution of this paper is in developing an efficient approach to investigate the reliability of AGVs taking into account the profiles of the mission undertaken.

The remaining part of the paper is organized as follows. In Section 2, the FMECA-FTA based methodology for AGV safety and reliability analysis is introduced; in Section 3, AGV risk and reliability analysis procedure is developed; in Section 4, the proposed methodology is applied to, assess an AGV's probability of success in completing a prescribed mission, identify the crucial phases of the mission and key AGV components. The work is finally concluded in Section 5 with concluding remarks and the plans for future work.

2 Overview of Methods and Application Area

FMECA and FTA are combined in this paper to develop a methodology for the safety and reliability analysis of an AGV system. To facilitate understanding, a brief introduction to both techniques is given below.

2.1 Failure Modes Effects and Criticality Analysis (FMECA)

FMECA originated from Failure Modes and Effects Analysis (FMEA)which is a well known popular technique used for dealing with safety and reliability issues in complex systems, such as identifying the potential effects that might arise from malfunctions of military, aeronautics and aerospace systems [7]. FMEA can be also used to implement the analysis of component failure modes, their resultant effects and secondary influences on both local component function and the performance of the whole system. A more detailed description of FMEA can be found in the standard [8]. In engineering practice, FMEA is often implemented at the early stage of system development such that the critical system components and potential failures and risks can be identified early.

Conventional FMEA covers the comprehensive analysis of components or subsystem, failure modes, and failure effects which are local and related to the overall system. FMEA can be further extended to rank the failure modes according to their probabilities of failure

and severity classifications based on the available data. That is the so-called Failure Modes Effect and Criticality Analysis (FMECA), where, 'criticality' is a terminology used to reflect the combined impact of 'occurrence probability' and 'severity' on the safety and reliability of the system being inspected.

2.2 Fault Tree Analysis (FTA)

Through inspecting the logic between the undesired events that could happen in a system or a mission, FTA allows us to trace back the root cause of a system or mission failure by using a systematic top-down approach. Moreover, the probability of system or mission failure can be computed via Boolean algebra from the tree. FTA provides a straightforward and clear presentation of the logic between various undesired events and is regarded as an effective, systematic, accurate and predictive method to deal with the safety and reliability problems in complex systems, such as the safety issues in a nuclear power plant [9]. In terms of structure, a fault tree is basically composed of various events and gates. In this paper, three basic types of gates, i.e. AND, OR and NOT, are used to depict the logical relations between the events that result in the occurrence of a higher level event. A more detailed description of FTA can be found in [7].

Since FTA ignores the cause of failure modes and FMECA gears towards analysing individual component failure mode occurrences and moreover it is not quantitative, both methods are used in combination in this paper.

2.3 Application AGV System

In this paper, to facilitate the research a typical AGV system is chosen for FMECA-FTA analysis. The AGV system consists of a laser navigation system, safety system, manual button, batteries, AGV software control system (ASCS), drive unit, brake system, steering system and attachments. Among these subassemblies, the laser navigation system, developed by Macleod et al. [10], is in essence a position measurement system to locate the AGV. It comprises a rotating laser installed on the board of the AGV and three beacons mounted along the border of the area to be covered. The safety system, with the aid of a laser detection system installed on the AGV, is designed to avoid obstacles that could appear on the pathway. These, together with the button, input into the control system. This system will then use the information to send commands to the drive unit, brake system, steering system and attachments, where the drive unit, usually a brushless DC electric motor, provides power for motion and operation. Attachments refer to those additional components that are used to assist moving and carrying items and batteries, usually the common lead–acid batteries, are used to supply power to the whole AGV system, see Figure 1.

The AGV that is studied in this paper is required to distribute materials to multiple places in a warehouse according to different requirements. Each time it receives an order it will optimise the routes for completing the whole mission first. Then, the AGV will travel to the material collection port along the optimised route to pick up the materials. After the AGV is loaded, it will travel to the storage station and unload the materials. After successfully distributing the materials, the AGV will travel back to its original parking position. Therefore, the whole mission can be divided into 6 phases in total, namely (1) mission allocation and route optimisation, (2) dispatch to station, (3) loading of item, (4) travelling to storage, (5) unloading and finally (6) travelling back to base. The mission can be regarded as successful only when the AGV is able to operate successfully throughout all these 6 phases without any break due to component and/or system failures and maintenance. Such a period is named as a maintenance-free operational period (MFOP) [11].

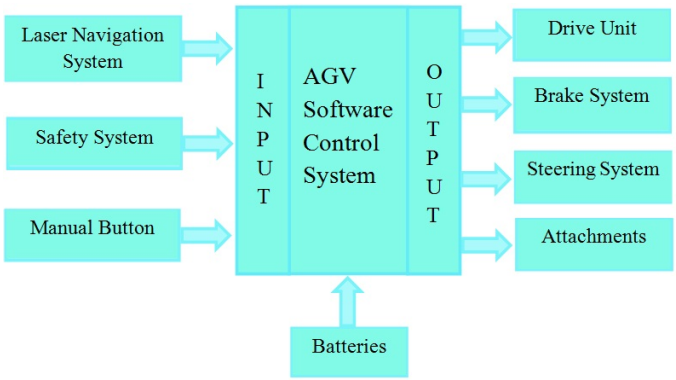

Figure 1 AGV system schematic.

Table 1 Severity assessment.

S_i	Description
1	No loss of any kind
2	Minor property loss (low cost hardware parts), no effect on performance
3	Major property loss, degradation of item functional output
4	Loss of critical hardware, human injuries, severe reduction of functional performance
5	Catastrophic loss of life, loss of the entire AGV system, serious environment damage

3 AGV Risk and Reliability Analysis Procedure

Applying the FMECA process requires the identification of the failure modes of all components in the AGV system, assessment of their local and system effects, evaluation of the severities of their consequences, and carrying out the analysis of their failure rates. The end outcome is the identification of a number of critical components in the AGV system based on the following risk priority number (RPN), i.e.

$$RPN_i = S_i \times F_i \times D_i \qquad (i = 1, 2, \ldots, N) \tag{1}$$

where N refers to the total number of failure modes of the AGV components being considered; S_i, F_i and D_i are the severity level, failure frequency and detectability of the failure of the i-th failure mode, respectively. In principle, the larger the value of RPN, the more critical (or important) the corresponding failure mode of the AGV component tends to be. In the calculation, the severity level S_i is assessed using the method depicted in Table 1. The failure frequency F_i is assessed based on the ranges listed in Table 2. The detectability D_i is assessed based on the information described in Table 3.

Once the critical components and their failure modes in the AGV system are identified by using the aforementioned FMECA method, the logical relations of the failure events of these identified critical AGV components will be investigated by the approach of FTA. The resultant fault tree can be constructed by using the following method:

(1) three basic logical gates, i.e. AND, OR and NOT, are used to depict the logical relations between the failure events that result in the occurrence of a higher level event;
(2) mission failure is used as the top event in the fault tree;
(3) phase failures and component/subsystem failures are used as the intermediate events;
(4) the various failure modes that lead to intermediate events are the fundamental events in the fault tree.

Table 2 Failure frequency assessment.

Failure frequency F_i	Range
1	$<$ 0.01 failures/ year
2	0.01-0.1 failures/year
3	0.1-0.5 failures/year
4	0.5-1 failures/year
5	$>$1 failures/year

Table 3 Detectability assessment.

Detectability D_i	Description
1	Almost certain to detect
2	Good chance of detecting
3	May not detect
4	Unlikely to detect
5	Very unlikely to detect

Table 4 Assumed phase lengths.

Phase	1	2	3	4	5	6
Phase Length (minutes)	1.2	12	1.2	9	1.2	6

Herein, it is necessary to note that the following two basic assumptions are presumed in the modelling process in order to simplify the topology structure of the fault tree and therefore model calculations:

(1) the AGV is presumed not to be assigned another mission after unloading; and

(2) the interactions between multiple AGVs, such as AGVs collision and deadlock, are neglected.

Once the fault tree is obtained, the phase unreliability, the failure probability of each AGV subsystem during the period of completing a prescribed mission, and the probability that the AGV is able to complete the whole mission, can be calculated through performing FTA. Hence, the safety, reliability and availability of the AGV system can be readily obtained.

4 Validation of the proposed methodology

In order to validate the methodology proposed in Section 3, the method is applied to identify the key AGV components (by FMECA), crucial phases of mission (by FTA), and assess the AGV's probability of success in completing a prescribed mission (by FTA).

To implement FMECA and FTA, the length (i.e. time duration) of each phase identified in Section 2.3, is prescribed a value as shown in Table 4. The total time duration to complete the whole mission is 30.6 minutes. It is worth noting that the data presented in Table 4 is empirical data, only for demonstration purpose. In reality, these would be different when the AGV implements different types of missions.

A list of the different failure modes of all the components of the AGV of interest are given in Table 5, in which the corresponding severity, failure rate, and detectability are included. The data in the table is based on [12] and expert knowledge and further developed for the AGV. The failure frequencies are given in number of failures per year. The resultant RPN can be determined from equation (1) and hence the criticality of each event failure mode can be ranked.

In the RPN calculation, the failure frequency F_i of each failure mode is derived first from the failure rate listed in Table 5 and the ranges described in Table 2. In table 5, the description of the component function, local and system effects and criticality ranking is not included due to limited space. Once the RPN of a failure mode is obtained, its criticality can be ranked.

From the FMECA results shown in Table 5, it can be found that the manual button has the smallest RPN and hence the lowest rank of criticality. Thus, it will not be regarded

Table 5 FMECA of AGV.

Identity	Sub-item	Failure Mode	S_i	F Rate(f/y)	D_i	RPN
Drive Unit		Unit fails	3	1	1	12
		Circuit connection fails	3	0.5	3	27
ASCS		Control system fails	3	2	4	60
		Control system malfunction	4	4	5	100
LNS	GPS	Fail to locate AGV	3	0.25	4	36
	Transmitter	Disabled communication	3	0.25	4	36
	Laser emitter	Unit fails	3	0.25	4	36
	Laser sensor	Unit fails	2	0.125	4	24
Safety Systems	Laser emitter	Unit fails	3	0.25	4	36
	Laser sensor	Unit fails	4	0.125	4	48
Attachments	Transfer part	Worn, fatigue, Looseness	4	1	2	32
	Holding part	Worn, fatigue, Looseness	4	1	2	32
Batteries		Performance degeneration	2	1	3	24
		Leakage	5	0.125	2	30
		Overheat	5	0.125	1	15
Brake System	Brake shoe	Worn out; Looseness	4	0.2	2	24
Steering System		Unit Fails	3	0.25	4	36
Manual button		Button is stuck	2	0.05	2	8

as key component of the AGV in the process of fault tree construction. Accordingly, for simplicity the fault tree of the AGV is built by only considering those identified key AGV components and the phases that they are involved in, i.e. drive unit, ASCS, laser navigation system, safety system, attachments, batteries, brake system, and steering system.

The construction of fault trees for phased missions is started by identifying the logic of different phases and their effects on the success of mission. Thus, 'mission failure' is chosen as the top event, and the 6 phases defined in Section 2.3 are used as intermediate event below this. The logic between the top event and these branch events is shown in Figure 2. The fault tree is further developed in order to investigate the logic between every phase mission and the failure modes of related AGV components. The resultant fault tree for Phase 2 is shown in Figure 3 as example.

From Figure 3, it is seen that the failure during the phase is used as the top event, the failures of those AGV components that are involved in the phase are basic events. The failures of mechanical parts, system parts for navigation, control and safety, and the power supply are the intermediate events. For example, in Phase 2, the AGV will travel from its parking position to the material collection port. During this period, the ASCS will control the AGV to travel along the optimised route; the laser navigation system (LNS) works over the whole course of the phase to locate the AGV as it moves; the motor is required to drive the vehicle; steering system enables vehicle turning; the safety system performs obstacle scan; and the brake system is responsible to slow down the vehicle when turning and stop the vehicle to avoid collisions. Obviously, the success of phase 2 mission relies on all of these subassemblies working. The fault of anyone of them can lead to the failure of phase 2. In addition, phase 2 can be started only after phase 1 has been completed successfully. In other words, the mission failure in phase $j+1$ is the combined result of successful phases 1 to j and the system failure occurring in phase $j+1$ via an 'AND' gate. This can be seen in Figure 3 where the 'NOT' gate is used to represent system success during phase 1 as NOT failure in

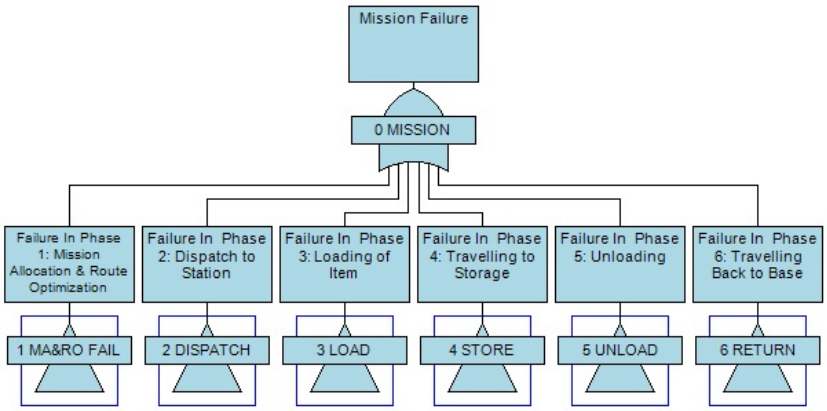

Figure 2 Logic between the top mishap and branch events.

Table 6 Component failures causing system failure at each phase

Phase	Component failures causing system failure at each phase
1	ASCS; LNS; Batteries
2	Drive unit; Brake system; Steering system; ASCS; LNS; Safety system; batteries
3	Attachments; Brake system; ASCS; Safety system; Batteries;
4	Drive unit; ASCS; LNS; Safety system; Attachments; Batteries; Brake system; Steering system
5	Attachments; Brake system; ASCS; Safety system; Batteries
6	Drive unit; ASCS; LNS; Safety system; Batteries; Brake system; Steering system

phase 1. Following this logic, the AGV operation is analysed at each phase. The component failures resulting in the system failure at different phases are identified as shown in Table 6.

Furthermore, in order to complete the FTA, the fault trees for all the identified critical AGV subsystems are further constructed. The corresponding fault trees for the drive unit and the ASCS are shown in Figure 4 as examples.

As a systemic FTA method has been developed in [13] dedicated to modelling phased mission with MFOP, that method is used in this paper to calculate the mission reliability and phased unreliability of the AGV within MFOP based on the phase lengths assumed in Table 4 and the FMECA information obtained in Table 5. The details of the calculation method are given below.

Firstly, the system failure in phase j, i.e. T_j, is calculated by using the following equation

$$T_j = (Phase\ 1\ to\ j-1\ Success).(Phase\ j\ Failure) \tag{2}$$

The probability of failure of basic event A in all phases from i to j (i.e. $q_{A_{i,j}}$) can be calculated using the equation

$$q_{A_{i,j}} = e^{-\lambda_A t_{i-1}} - e^{-\lambda_A t_j} \tag{3}$$

where λ_A refers to the failure rate of a basic event A, t_j is the length of phase j.

The unreliability of phase j can be calculated by

$$Q_j = 1 - R_j = 1 - R_{1,j}/R_{1,j-1} \tag{4}$$

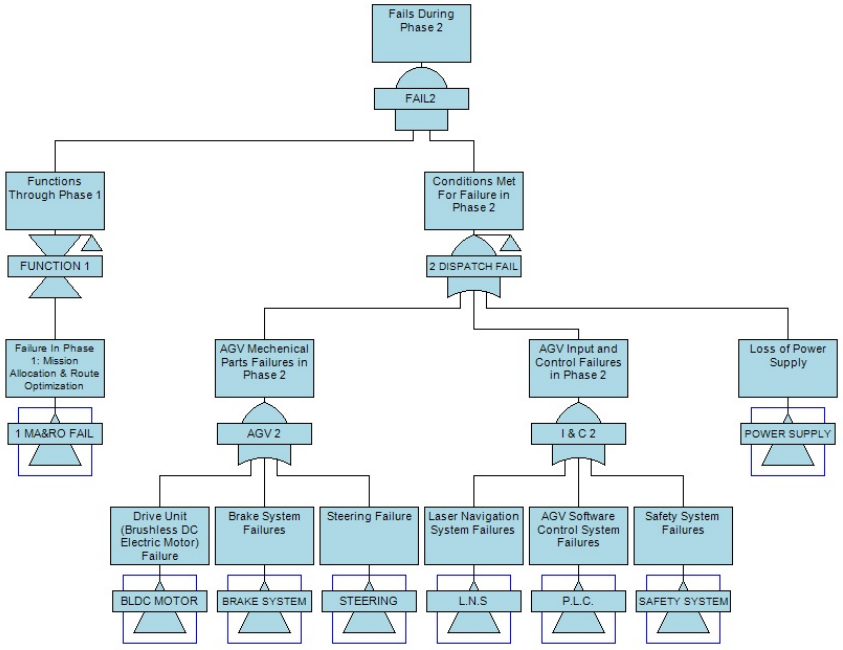

Figure 3 Fault trees for Phase 2.

Table 7 Component failure probability at the end of whole mission.

Description	Failure Probability	Description	Failure Probability
ASCS	0.00034925	LNS	0.00005094
Attachments	0.00009360	Safety Systems	0.00002183
Drive Unit	0.00008725	Steering System	0.00001455
Batteries	0.00007277	Brake System	0.00001164

where R_j denotes the success probability of phase j , $R_{1,j}$ is the success probability till the end of phase j. It should be noticed that the probability of failure is calculated using the exponential distribution since the failure rates are assumed to be constant for simplicity.

In the FTA calculation, the component will be taken into account in a phase only when it is involved in the completion of that phase. It will not be considered if it contributes nothing to the phase. Applying the aforementioned method to calculate the component failure probability, mission reliability, and phased unreliability of the AGV within MFOP, the results are shown in Tables 7 and 8.

From the results shown in Table 7, it is seen that the ASCS, attachments, drive unit and battery have the largest failure probability at the end of the whole mission. That implies these four components are most vulnerable to failure.

From Table 8, it is found that the mission reliability at the end of the 6th phase is 0.99930, which is based on the success of all six phases. This means that the AGV has a greater than 99% chance of successfully completing the mission. Thus, it in fact indicates the overall reliability of the AGV in accomplishing the whole mission. For this reason, it can be concluded that the AGV considered here is a very reliable material distribution vehicle in the warehouse. In addition, Table 8 shows that phase 2 ‘dispatch to station’ and phase 4 ‘travelling to storage” show the largest phase unreliability values. This means that the AGV is more likely to fail in the completion of these two phases.

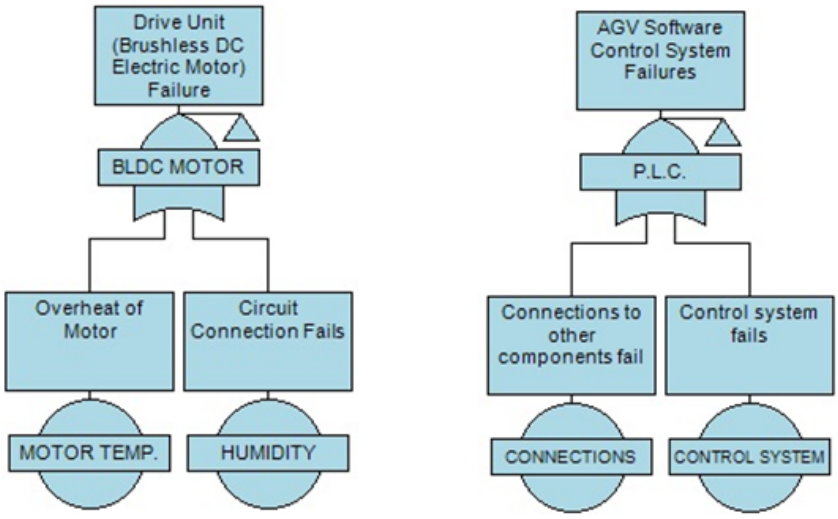

Figure 4 Fault trees for Drive Unit and ASCS.

Table 8 The resultant mission reliability and phase unreliability.

Phase	Mission reliability at phase end	Phase unreliability
1	0.99998	0.00001855
2	0.99974	0.00024386
3	0.99967	0.00007266
4	0.99945	0.00021915
5	0.99942	0.00002243
6	0.99930	0.00012527

Additionally, in this paper the optimum maintenance time of the AGV is also considered by investigating the reliability of the AGV against the number of the missions that the AGV can complete without maintenance. The success probability of the AGV can be calculated by

$$SuccessProbability = \prod_{i=1}^{n} P_i \tag{5}$$

where P_i is the probability that the AGV is able to complete the i^{th} mission successfully, and n is the number of missions the AGV needs to complete. The calculation results are shown in Figure 5. From Figure 5, it is interestingly found that with the increase of the number of missions that the AGV can complete without receiving any maintenance, the success probability shows a monotonous decreasing tendency, thus from which the optimum maintenance time of the AGV can be readily inferred.

5 Conclusions

In order to investigate the safety and reliability issues existing in AGVs that are being increasingly used for intelligent transportation and material distribution in warehouses and/or manufacturing facilities, a promising technical approach has been established in this paper. It has been shown that FMECA and FTA can be adopted to identify the critical

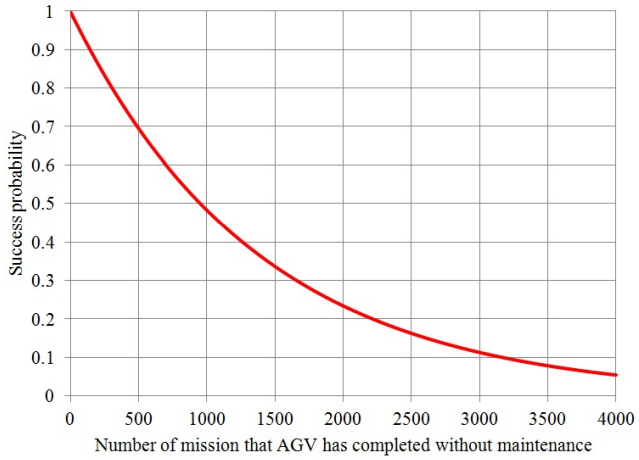

Figure 5 Success probability vs. mission number.

AGV components and the crucial mission phases of AGVs at the design stage. From the research reported, the following conclusions can be reached:

(1) The key AGV components can be successfully identified based on the criticality rank that is obtained through performing FMECA. The calculation results presented in this paper has shown that nearly all AGV components except manual button, such as driving, operating, control and power supply units, are critical components;

(2) The FTA results show that among all identified key components, the ASCS, attachments, drive unit and battery are most vulnerable components to failure because they are found to have the largest failure probability at the end of whole mission;

(3) The FTA calculation has suggested that the AGV is more likely to fail in the completion of the phase 'dispatch to station' and the phase 'travelling to storage' because these two phases show the largest phase unreliability values. But it is worth noting that such conclusions are based on the assumptions given in Section 3. In reality, the result would be different, depending on the real reliability data collected from the AGVs;

(4) Research has shown that the AGV being inspected is overall a very reliable material distribution vehicle in the warehouse. But Figure 5 has indicated that the reliability of the AGV will degenerate if it completes more missions without maintenance;

(5) Through this research, it can be concluded that the proposed FMECA-FTA approach is indeed a valid method for assessing and evaluating the safety and reliability issues in AGVs.

Nevertheless, the work reported in this paper is only a preliminary research on AGV reliability issues. In the future, the proposed method will be further validated by using real AGV data through the collaboration with relevant industry partners.

Acknowledgements. The work reported in this paper supports the research in the grant by EPSRC – project titled Adaptive Informatics for Intelligent Manufacturing(AI2M). The authors would like to say thanks to Dr. Cunjia Liu at Loughborough University and Mr. Dave Berridge at Automated Materials Handling Systems Association for their kind help in the preparation of this work.

References

1 Müller T. *Automated Guided Vehicles*. IFS (Publications) Ltd./Springer-Verlag, UK/Berlin, 1983.

2 Iris F. A. Vis *Survey of research in the design and control of automated guided vehicle systems*. European Journal of Operational Research, Vol. 170, Issue 3, pp. 677–709, 2006.

3 Giuseppe C. and Marcello F. *A network flow based heuristic approach for optimising AGV movements*. J IntellManuf, Vol. 24, pp. 405–419, 2013.

4 Wu N. Q. and Zhou M. C. *Zone Design of Tandem Loop AGVs Path with Hybrid Algorithm*. IEEE/ASME Transactions on Mechatronics, Vol. 12, No. 1, pp. 63–72, 2007.

5 Fazlollahtabar H. and Saidi-Mehrabad M. *Optimising a multi-objective reliability assessment in multiple AGV manufacturing system*. International journal of services and operations management, Vol. 16, No. 3, pp. 352–372, 2013.

6 Duran D. R., Robinson E. Kornecki A.J. and Zalewski J. *Safety Analysis of Autonomous Ground Vehicle Optical Systems: Bayesian Belief Networks Approach*.2013 Federated Conference on Computer Science and Information Systems, pp. 1419–1425, 2013.

7 Andrews J. D. and Moss T. R. *Reliability and Risk Assessment (second Edition)*, Professional Engineering Publishing, 2002.

8 MIL-STD-1629A (1980); and MIL-STD-1629 Notice 2, 1984.

9 U.S. Nuclear Regulatory Commission. *Severe Accident Risks: An Assessment for Five U.S. Nuclear Power Plants – Final Summary Report (NUREG-1150, Volume 1) Introductory Pages and Part I – Chapters 1 and 2*. Office of Nuclear Regulatory Research, Vol. 1, 1990.

10 MacLeod E. N. and Chiarella M. *Navigation and control breakthrough for automated mobility*. In: Proceedings of the SPIE mobile robotics VIII, Boston, MA, 1993.

11 Hockley C. J. *Design for success*, Proc Inst Mech Eng 212(G):371–8, 1998.

12 FMD-97 *Failure Mode/Mechanism Distributions*. December 1997.

13 Chew S. P., Dunnett S. J. and Andrews J. D. *Phased mission modelling of systems with maintenance-free operating periods using simulated Petri nets*. Reliab Eng Syst Safe Vol. 93, No. 7, pp. 980–994, 2008.

Petri Net Modelling for Enhanced IT Asset Recycling Solutions

Christina Latsou[1], Sarah J. Dunnett[2], and Lisa M. Jackson[3]

1 **Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, Leicestershire LE11 3TU, UK**
C.Latsou@lboro.ac.uk
2 **Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, Leicestershire LE11 3TU, UK**
3 **Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, Leicestershire LE11 3TU, UK**

Abstract

From preliminary design through product sustainment to end of life removal, optimal performance through the entire life cycle, is one of the most important design considerations in engineering systems. There are a number of mathematical modelling techniques available to determine the performance of any system, or process design. This paper focuses on the Petri Net technique for the representation and simulation of complex cases with the future aim of automatically generating a model from the system, or process description. If the model can be automatically generated changes can be investigated easily, enabling different designs to be investigated. Within this research, a Petri Net model is developed for a process of recycling IT assets. The model developed here will be used in future work to validate the automation process. This model is simulated and programmed in Matlab. The model enables the simulation of various flow paths through the recycling process, giving an understanding of the current process limiting factors. These can then be used to identify possible ways of improving the efficiency of the recycling process and enhancing the current IT asset management strategy. The future aim of this research is the automatic generation of a system model for complex industrial systems and processes by converting the SysML-based specifications into Petri Nets.

1998 ACM Subject Classification D.2.2 Design Tools and Techniques, I.6 Simulation and Modelling

Keywords and phrases Petri Net, simulation, modelling, asset management

Digital Object Identifier 10.4230/OASIcs.SCOR.2016.3

1 Introduction

At the earliest stages of the design of a system, or process, it is essential to consider its performance in order to ensure that the optimal design is developed. A number of modelling techniques are available, facilitating the performance evaluation and analysis of any system, or process. Once the performance is understood, the decision maker is able to make informed decisions about the design. The techniques can be divided into two main categories, analytical and simulation techniques. The analytical techniques include a wide variety of approaches. An analyst is able to choose the most suitable technique, given the available data and the preferred analysis that needs to be performed. The techniques for failure analysis consist of combinatorial models, including Reliability Block Diagrams (RBDs), Fault Trees (FTs) and Binary Decision Diagrams (BDDs), state-space models, including the subcategory of Markov approaches, and hierarchical models generated by the combination of combinatorial

and state-space models, which are able to simplify the model and ease further analysis [1]. In the case of investigating other performance measures, different analytical techniques can be used. The models mentioned above contain certain limitations rendering them not applicable to some cases, especially when the system, or process, includes complex structures. More specifically, the combinatorial models are not able to model dynamic characteristics, such as dependent events and spares, whereas the Markov models, which can cope with dynamic features, suffer the state-space explosion as the number of states increases. In other words, large complex scenarios are difficult to be modelled and controlled using Markov approaches, as the final diagram can be very large, difficult to be built due to its complexity and computationally costly.

However, alternative approaches have been developed, such as the encoding of the state-space model in a Petri Net (PN) that can cope with all the aforementioned limitations [7]. Petri Net models are powerful, flexible structures that can be applied to complex cases without suffering the state-space explosion limitation.

The main characteristics of a complex case are redundancy, configurability and interactions between systems components or processes scenarios. These features are usually incorporated into the system, or process creating high level of difficulties in the manual development process of performance models. Complex cases can be recognized by the following factors: the system size (number of different components, or cases within it), the system dynamics i.e. the extent to which behaviour changes over time, the fault and error recovery, the sequence-dependent failures, the use of spares and the existence of phased missions [4].

Nowadays, the use of computational models is considered an essential element for the analysis of complex cases. This paper uses the simulation technique which has become a ubiquitous tool for the analysis and investigation of complex cases. It is characterised as a straightforward, accurate, easily applicable and flexible method that can represent the system, or process in an efficient way and hence it can provide reusability of the model for different designs and policies, enabling informed decisions to be made. Hence, in this paper a simulation method is generated with the modelling power of Petri Nets to model a process of recycling IT assets. The results from the simulation of the various flow paths of the process give the opportunity for the analyst to detect the limiting factors in the process and make the appropriate decisions. As a result, the efficiency of the current process can be improved and the IT asset management strategy can be enhanced.

The future aim of the project is development of a general methodology for complex cases, implemented in computer software, which will accept as input a system, or process description and automatically generate the corresponding Petri Net model. The study developed here is the first step in this process, the manual development of a PN. The future stages will develop an algorithm to do this automatically for the process described. This will be generalised for any complex system, or process. This paper is organised as follows. Section 2 describes briefly the PNs giving a general overview of basic concepts and elements of the models. In Section 3, a PN model is developed for a process of recycling IT assets and results are obtained from the net highlighting certain aspects of the process. Some general conclusions are drawn in Section 4.

2 Petri Nets

First introduced in the thesis of C. A. Petri in 1962 [5], Petri Net is a visual tool that provides rigorous and precise model analysis. PNs have been applied to a wide spectrum of cases in different sectors, such as data communication processes, computer networks, workflows and

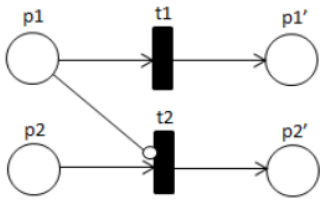

Figure 1 Inhibitor Arc.

manufacturing plants [6]. A Petri Net is a bipartite directed graph that includes two types of nodes: places, drawn as circles, and transitions, drawn as bars. There are two types of transitions, the immediate (drawn as solid rectangles) which when enabled fire immediately, and timed (drawn as hollow bars) which have a time delay associated with them. Directed edges (arcs) connect places to transitions and vice versa. The dynamic behaviour of the model is described by the movement of tokens/markers (solid dots) between places. A token can move between places only if the corresponding transition has been enabled. A transition is said to be enabled and able to fire if the number of tokens in each of its input places is at least equal to the multiplicity of the corresponding arc from that place. The multiplicity of an arc is described as the number of the tokens that are removed from the place during a firing transition. This multiplicity is denoted by using a slash through the arc a positive integer beside it. When the multiplicity is equal to 1, then the slash and integer can be omitted. Another element that was added in order to increase the decision power of the Petri Nets is the inhibitor arc, denoted as an arc terminated with a hollow circle. The inhibitor arc prevents the firing of a transition when the place it comes from is marked. According to Figure 1 t1 is enabled if p1 contains a token, while t2 is enabled if p2 contains a token and p1 has no token.

The movement of tokens through a Petri Net can be transformed into matrix form. Then the marking of the PN after the $_r$ transition, M_r, can be found by Equation (1).

$$M_r = M_0 + A^T \cdot T_1 \,. \tag{1}$$

Where: M_0 is a column matrix $(n, 1)$, where n is the number of places, showing the initial marking of the net. T_1 is a column matrix $(m, 1)$ where m is the number of transitions, showing the number of times each transition has fired in the r transitions. A is the incidence matrix (m, n) where each element a_{ij} corresponds to the effect that transition i has on place j. Using Equation (1), the marking of a net, the distribution of tokens within it, can be determined at any time.

Petri Net models can predict the performance of complex processes, being a powerful modelling tool suitable for the integration of continuous and discrete dynamics in the model. According to Ling [2] reachability, boundedness and liveness are three main behaviour properties that relate to each other and can be identified in PN models. Reachability is the capability to reach one particular state (M_0) from another (M_0') if there is a sequence of transitions such that $M_0 \leq M_0'$. A Petri Net is characterised as bounded if the number of tokens in each place is less or equal to a finite number k for any marking reachable from the initial marking. The term liveness ensures that a transition is live if it can never deadlock.

In the area of applied mathematics Petri Nets are of special interest since they can be analysed, simulated and modelled numerically and graphically, using various software tools, such as C++, Matlab/Simulink, PN Toolbox [3], SimHPN (GISED) and their tools.

In this application, given the IT asset process, a Petri Net model is developed. The novelty of application in this domain for asset management of recycling is the ability to investigate

and identify possible limiting factors in the process and hence optimise the performance of the IT process.

3 Simulation

The Petri Net technique, described above, can be used for the simulation of a process in order to conduct performance analysis. To demonstrate the technique of simulation, a PN model has been developed for a process of recycling IT assets. The model enables the simulation of various flow paths through the recycling process. The simulation results will give the ability to the analyst to identify possible ways of improving the performance of the process and enhancing the current IT asset management strategy.

3.1 Process Description

A recycling IT process has been considered and modelled using Petri Nets in order to simulate its performance. The process focuses on the repair of electronic devices, primarily mobiles phones. The product enters in the process line and then it can pass along one of the two paths, either refurbished, through the reuse line, or scrap. Considering the refurbished path, there are seven different possible stages:

- Asset Track (AT): Asset information is introduced into the traceability system.
- Visual Inspection (VI): The physical condition of an asset is assessed.
- Functional Test (FT): A product is inspected by testing its functionality, including activities such as charger check, battery test, LCD screen check, resetting, ringing test, vibration, microphone and speaker test.
- Data Erasure (DE): Data is erased securely by using specific licensed software.
- Cleaning and De-Labelling (C&D): Refurbished assets are cleaned and any non-essential labels are removed from the device. A new label is placed.
- Repair (R): A product can be repaired only if its repair is considered economically viable.
- Strip and Scrap (S&S): Failed products are checked for any parts that can be salvaged and are then sent for secure destruction.

In the case of a scrapped device, there are two options for it. It can be used either as a unit level, meaning price sought per tonne for scrap, or as a component level, meaning components are extracted from the device and used within this process for future repairs or sold for spares. All stages can only have one device at a time apart from the Data Erasure stage. Each stage has a time to completion associated with it, which can vary for different devices. Additionally, at some of the stage there is a probability of pass/ fail, according to the process. In practise, most of the activities are performed at the same physical location, i.e. on the computer. The repair stage (R) takes place in the same factory, but it is performed when there are a batch requiring repair and it takes place away from the main refurbishment process and for that reason there is a large delay between the functional test and the repair activities. Similarly, the data erasure stage is performed separately and then the information is only logged in the process at the end. The company's manpower is assumed to be 2.

The process diagram is presented in Figure 2, including all the possible paths of the recycling IT process.

3.2 Model Construction

From the information given, as described in Section 3.1, the corresponding Petri Net model has been developed and presented in Figure 3. The PN model includes all the stages referred

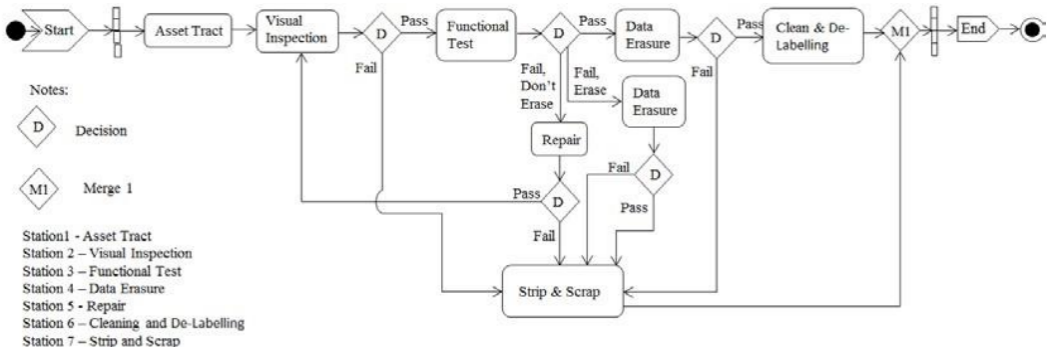

Figure 2 Recycling IT Process Diagram.

to in the process description. Two places and a timed transition have been used to represent each activity, i.e. one place is used to describe that the activity starts, whereas a second place is used to describe that the activity ends. The time for the activity to be completed is described by a timed transition; these are denoted as Tx, where x is a number (1–8), in Figure 3. It can be seen that there is a TI node which is used as a merge place only if two or more devices arrive at the same place. Inhibitors have been placed in the PN, as shown in Figure 3, allowing the firing of a transition only if the subsequent place is empty. This models the queuing of items. There is an immediate transition in the model, named TI and represented by a black box, which corresponds to an instantaneous transition.

Some stages of the process have a probability associated with their pass or fail and these are shown by probability transitions in Figure 3 denoted by Px, where x is a number (1–14). For example, at the start of the process after asset tracking has taken place a visual inspection occurs. This takes a time T2 and there is a probability P3 that the unit passes this inspection, etc. In order to model the time between stages in the model, places Pix, with associated times Tix, have been added. Where, as before, x is a number (1–33). So, in total the PN includes 29 places, 1 immediate transition and 33 timed transitions from which 8 correspond to the activities, 14 represent the intermediate time from one activity to another and the rest 11 transitions express the probabilities (pass/fail) of each activity.

3.3 Process Data and Model Simulation

In order to simulate the process modelled in Figure 3 it is necessary to input values for the timed and probabilistic transitions. Table 1 presents the minimum and maximum times needed to complete each activity and Table 2 shows the same data for the times between stages. The probabilities of pass (P) or fail (F) for each stage of the process are also listed in Table 2. The retrieved data, which was obtained from the IT asset management company, comes from 2113 mobile phones which have been processed for the time period of 323 hours.

Hence in the simulation, times are generated for the timed transitions by assuming that they all follow a continuous uniform distribution. For each transition a random number, x, is generated and the time obtained from Equation (2).

$$t = \min_t + [(\max_t - \min_t) \cdot x] \,. \tag{2}$$

Where:

- $\min_t$ and $\max_t$ are obtained from Tables 1 and 2 for each activity type and correspond to Min Time and Max Time respectively.
- x returns a uniformly distributed random number in the interval $(0, 1)$.

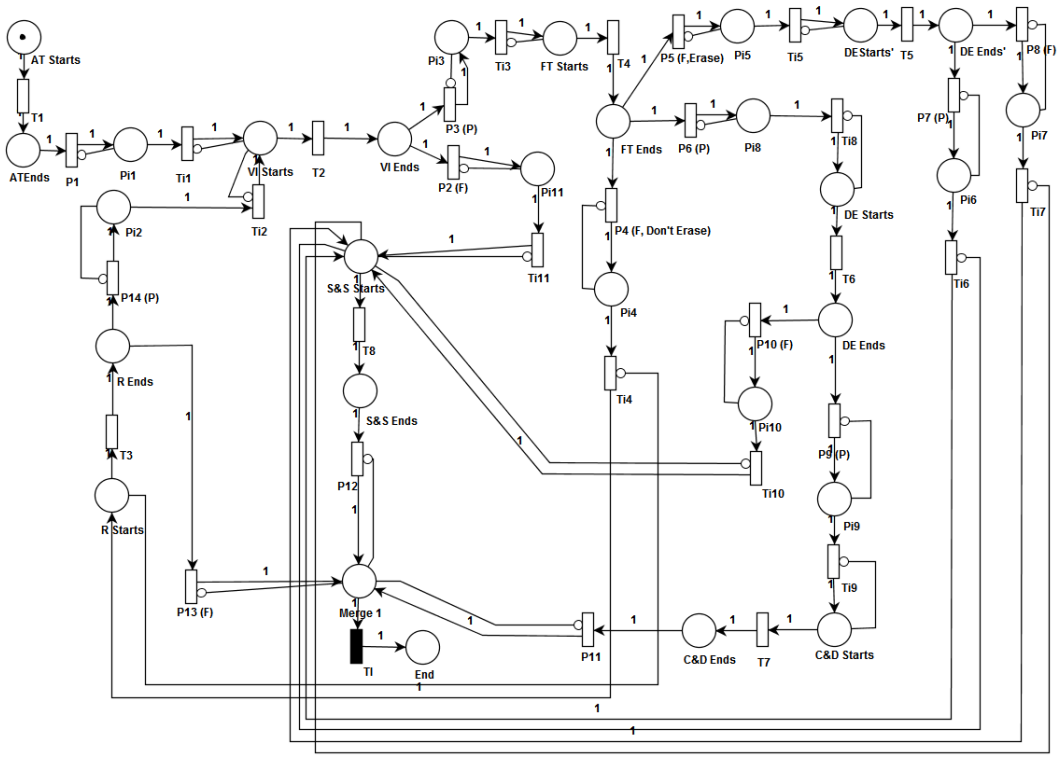

Figure 3 Recycling IT Process Petri Net Model.

The main steps followed for the process simulation are outlined in Figure 4. The Matlab code was generated and run for 1000 simulations. As mentioned, random numbers for the process times and probabilities are generated and times for activities and intermediate stages are estimated. Once a random probability is lower than or equal to the pass probability of an activity then the device is assumed to pass, otherwise the device fails. Following Figure 3 and the probabilities for each path the simulation results can be found. The simulation can provide the total number of repaired and scrapped devices, the average time for each activity and intermediate stage, the longest queues once more than one device exists in the process, as well as the most visited places and the most failed nodes in each path.

3.4 Results

This section presents the results obtained from the simulation of the PN developed in Section 3.2. The main focus of this application is the investigation of the process and identification of its limitations with the aim to find ways to enhance the performance of the process.

Table 3 presents the average times taken for the transitions included in the repair path for the 1000 simulations undertaken. It is observed that Ti4, which is the time needed from the end of the FT to the start of the R activity, is the longest time in this path. This happens because the r activity takes place in a different location from the rest of the activities and extra time is required for the transportation of the devices. As can be seen that the average times obtained agree well with the times given in Tables 1 and 2. According to the 6 paths identified in the PN and Table 5, the most common failed nodes have been found in each path

Table 1 Average Times Table for Process Stages.

Timed Transitions	Activity Type	Min Time (seconds)	Max Time (seconds)
T1	Asset Track	107	148
T2	Visual Inspection	5	10
T3	Repair	240	900
T4	Functional Test	60	180
T5, T6	Data Erasure	30	40
T7	Cleaning and De-Labelling	30	60
T8	Strip and Scrap	30	60

Table 2 Times between Stages and Probabilities for the IT Process.

Transition ID	Min Time (seconds)	Max Time (seconds)	Transition ID	Probability
Ti1	30	120	P1,11,12(P)	1
Ti2	1800	28800	P2(F)	0.312
Ti3	300	1800	P3(P)	0.688
Ti4	7200	86400	P4,5(F)	0.13365
Ti5	1800	7200	P6(P)	0.7327
Ti6	1800	10800	P7,9(P)	0.9702
Ti7	1800	10800	P8,10(F)	0.0298
Ti8	1800	7200	P13(F)	0.7066
Ti9	1800	10800	P14(P)	0.2934
Ti10	1800	10800		
Ti11	300	3600		

and presented in Table 4. The initial total number of runs is equal to 1000 and the results for the places depend on the probabilities of their equivalent transitions. The most common failed place to have been visited, Pi11, corresponds to VI failing and has a failure probability equal to 0.312. In addition, the most common visited places have been investigated in each path and it was found that the activities with the most visits are the AT and VI passing nodes. More specifically, the AT nodes are always visited, whereas the VI passing places are visited approximately at 68% in all paths.

Table 6 includes the overall average times for all PN paths for two different cases; the first case investigates only one device in the process, whereas the second case investigates two. For the estimation of the average repair time in the case of one device, the activity and intermediate times have been taken into consideration. In the case of two devices the methodology presented in Figure 5 has been followed. All the activities in the repair path can have only one device at a time, meaning that the 2^{nd} device can proceed with the AT, once the 1^{st} phone ends with the AT. The same happens for the R. Secondly, an additional time that should be included in the case of two mobile phones is the time that corresponds to A number, presented in Table 6. The A number corresponds to the time in which the 1^{st} device waits for the 2^{nd} to complete its FT. Once the 2^{nd} device completes this activity, both devices are transported to a pre-specified location for their R (T4). The 2^{nd} device starts its repair when the repair of the 1^{st} device ends. The repair path presents the highest average time for both cases. This is due to the time taken for the transition of the device(s) from the FT to R. The same procedure was applied for the other paths and the additional average times in the case of two devices in the process are shown in Table 6 for each path. The f

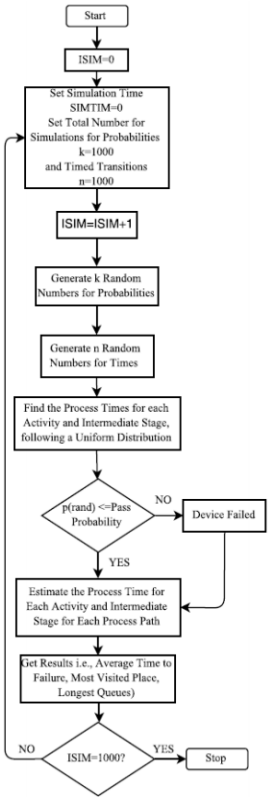

Figure 4 Flowchart for IT Process Simulation.

Table 3 Average Times of Timed Transitions in the Repair Process Path.

Timed Transitions ID	T1	Ti1	T2	Ti3	T4	Ti4	T3
Average Time (seconds)	127.77	74.951	7.510	1039.79	119.19	47536.15	569.968

numbers in the C&D and S&S paths correspond to the time that the 2^{nd} device waits for the 1^{st} to end with the T7 and T8 respectively. According to the A and f numbers, placed in Table 6, the longest queue has been identified in the repair path, since the sum of A and T4 is higher than the corresponding combinations of the other paths. The C&D and all the S&S paths have similar numbers of queues, since the f numbers are very close and T7 is equal to T8. Based on the simulation only the 2.69% of the devices can be repaired, the 48.9% undergoes the C&D and the rest of the devices lead to the S&S paths.

4 Conclusions

In this paper a process of recycling IT assets was investigated by developing and subsequently simulating a PN for the process. The simulation results have given a clear understanding of the working way for the six different paths of which the process consists and also main limitations and potential modifications have been identified in order to enhance the process performance. A main limiting factor of the process is the long time needed for the Repair path to be completed. This happens because the Repair activity takes place in a different physical location from the other activities. Additional observation for all the process paths is that the mobile phones spend the most time within the intermediate times rather than

Table 4 Most Common Failed Nodes in Each Process Path.

Node ID (Place)	Pi4	Pi3	Pi11	Pi5	Pi5	Pi8
No. of Failures	596.0384	312.0666	688.059	595.9192	596.0097	489.1843

Table 5 Paths in the IT Process.

Path	Activities
1	AT-VI(P)-FT(F-No-Erase)-R
2	AT-VI(P)-FT(P)-DE(P)-C&D
3	AT-VI(F)-S&S
4	AT-VI(P)-FT(F-Erase)-DE(P)-S&S
5	AT-VI(P)-FT(F-Erase)-DE(F)-S&S
6	AT-VI(P)-FT(P)-DE(F)-S&S

Table 6 Overall Average Times of Each Path for 1 and 2 Devises and Additional Time Required in the Case of 2 Devices.

Path	Avg. Time(1 phone)(secs)	Avg. Time(2 phones)(secs)	Avg. Additional Time(secs)	
1	4.87E+04	4.94E+04	A=126.7688,	T4
2	7763.5347	7941.2791	f=82.4264,	T7
3	2206.6686	2331.9461	f=82.4199,	T8
4	7761.9031	7940.2581	f=82.3125,	T8
5	7762.7325	7941.008	f=82.4972,	T8
6	7763.5059	7941.1462	f=82.9941,	T8

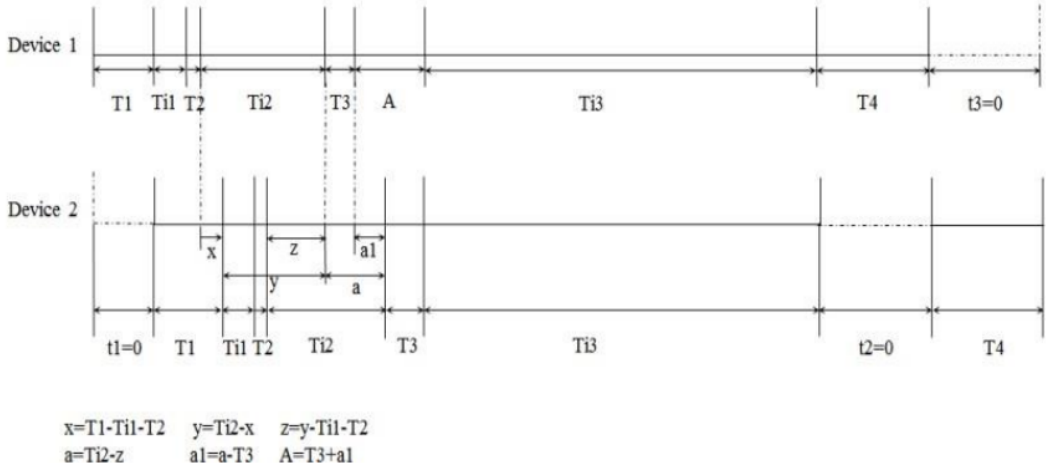

Figure 5 Estimation of Repair Average Time for 2 Devices.

within the activities. So, some potential ways of improving the efficiency of the process could be the duplication of the activities, increase of the manpower and decrease of the intermediate times, for example the devices could be repaired at the same location, or the transportation of the devices could be arranged to take place after the end of the workers' shifts and hence the intermediate time between the Functional Test and Repair would not affect the whole time process. Another recommendation for the process enhancement is the creation of standards for the activities to accept multiple devices simultaneously, so that this can save not only time and money but also can increase the reputation of the company.

The future work involves; (a) applying the proposed process to a variety of possible scenarios, i.e. more than two devices in the process. This will allow the user to identify where the technical investment should be made or how to rearrange the process in order to improve its performance and (b) developing an algorithm to generate automatically the PN model for the IT process from the system, or process description. This algorithm will then be generalised for any complex system, or process. The manual PN developed in this work will be used to validate the algorithm. The automated generation of the PN will contribute to the literature by enabling different designs to be investigated precisely, easily and quickly; hence allowing optimal designs to be determined.

References

1 Mark Lanus, Liang Yin, and Kishor S Trivedi. Hierarchical composition and aggregation of state-based availability and performability models. *IEEE Transactions on Reliability*, 52(1):44–52, 2003.

2 Chris Ling. An introduction to petri nets. [Power Point Presentation], 2001.

3 Christian Mahulea, Michaela H. Matcovschi, and Octavian Pastravanu. Home page of the petri net toolbox, 2003.

4 Seyed H. Nejad-Hosseinian. *Automatic generation of generalized event sequence diagrams for guiding simulation based dynamic probabilistic risk assessment of complex systems.* PhD thesis, University of Maryland, College Park, Department of Mechanical Engineering, 2007.

5 Carl A. Petri. *Communication with automata.* PhD thesis, Universität Hamburg, 1966.

6 Jiacun Wang. Petri nets for dynamic event-driven system modeling. *Handbook of Dynamic System Modeling*, pages 1–17, 2007.

7 Valérie Zille, Christophe Bérenguer, Antoine Grall, and Antoine Despujols. Simulation of maintained multicomponent systems for dependability assessment. In *Simulation Methods for Reliability and Availability of Complex Systems*, pages 253–272. Springer, 2010.

Simulation Combined Approach to Police Patrol Services Staffing

Hanjing Zhang[1], Antuela Tako[1], Lisa M. Jackson[3], and Jiyin Liu[4]

1 **School of Business and Economics, Loughborough University, Leicestershire, LE11 3TU, United Kingdom**
h.zhang@lboro.ac.uk
2 **School of Business and Economics, Loughborough University, Leicestershire, LE11 3TU, United Kingdom**
a.takou@lboro.ac.uk
3 **Aeronautical and Automotive Engineering, Loughborough University, Leicestershire, LE11 3TU, United Kingdom**
l.m.jackson@lboro.ac.uk
4 **School of Business and Economics, Loughborough University, Leicestershire, LE11 3TU, United Kingdom**
j.y.liu@lboro.ac.uk

Abstract

Motivated by the squeeze on public service expenditure, staffing is an important issue for service systems, which are required to maintain or even improve their service levels in order to meet general public demand. This paper considers Police Patrol Service Systems (PPSSs) where staffing issues are extremely serious and important because they have an impact on service costs, quality and public-safety. Police patrol service systems are of particularly interest because the demand for service exhibits large time-varying characteristics. In this case, incidents with different urgent grades have different targets of patrol officers' immediate attendances. A new method is proposed which aims to determine appropriate staffing levels. This method starts at a refinement of the Square Root Staffing (SRS) algorithm which introduces the possibility of a delay in responding to a priority incident. Simulation of queueing systems will then be implemented to indicate modifications in shift schedules. The proposed method is proved to be effective on a test instance generated from real patrol activity records in a local police force.

1998 ACM Subject Classification G.1.10 Applications

Keywords and phrases Police patrol service system, Time dependent queue, Priority queue, Square root staffing, Simulation

Digital Object Identifier 10.4230/OASIcs.SCOR.2016.4

1 Introduction

Performance measures of the Police Patrol Service Systems (PPSSs) typically focus on response times, especially immediate response to incidents which occupy a large proportion of police patrol resources. According to the nature of reported incidents, call handlers in front desks divide incidents into four grades related to their urgency. They are emergency response incidents, priority response incidents, scheduled response incidents and incidents that could be resolved without deployment. Only the first two grades of incidents, emergency response incidents with threat to life and priority response incidents with necessary officer attendance, require immediate response. The common target for PPSSs in the UK is to attend to over 85% of emergency response incidents within 15 minutes and reaching over

80% of priority calls in 60 minutes [4, 16]. Unlike the well scheduled demand in industry manufacturing and transportation crews in airlines, police emergency services providers always face time-dependent service requests. Patrol officers must be ready to attend any assigned incident of which the time and place cannot be known in advance. Managers need to make reasonable predictions of required staff levels based on historical records, and allocate enough patrol officers to avoid poor performance with long periods of under-staffing. Long periods of over-staffing should also be avoided with the objective being to minimise the workforce costs.

Response times, as an important performance measure for PPSSs, consist of pre-travel delays and travel times. Accurate prediction of system performance requires information of both parts. Travel times usually depend on the distance between the scene of the crime and the patrol officers' current locations. The difference in speed, caused by many factors including types of roads, traffic situations and incident grades, where for emergency incidents blue lights and sirens can be used, will also influence the time spent on travelling [13, 14]. It is a bit more complicated to analyse pre-travel delays since lack of staff is the main contributing factor. Using queueing systems with time-dependent arrival rates can model pre-travel delays in patrol service systems adequately, but it is not easy to analyse because most queueing theory focuses on long-run steady-state behaviour of stationary queueing models, rather than queueing models switching between under-staffing and over-staffing. Discrete event simulation provides explicit numerical results to evaluate the system performance. However, as simulation models are always built case by case, it is quite time consuming to exhaustively simulate all possible system configurations. In this paper, we propose a scheduling method which combines both analytical results and simulation results for time-dependent priority queueing model to indicate appropriate shift schedules. Comparing with traditional staffing methods, the proposed method adjusts varying staff levels into staff levels of shift patterns to cope with the demand fluctuations.

Traditional methods for solving staffing problems usually assume staff levels on different periods are independent from each other. Staff level is determined to satisfy the service target in each period at lowest cost. Then an integer programming model is solved which fits an optimal shift schedule to these staff levels. As a simple approach, pointwise stationary approximation [8] was proposed for setting staff levels in call centres. This approximation provides a basic analytic support to determine the staff level with piecewise stationary arrival and short service times. Queuing processes in real life are more complex since service time may be relatively long. As a modification, Square Root Staffing (SRS) rule [11] is proposed to capture dependency effects of different periods and more accurate estimations for staffing requirements can be expected. This method has been proved to provide demand estimations that are more thoroughly estimated. However, in two-step staffing algorithms, required staff levels determined in the first step are fixed for generating shift schedules in the second step. Several solutions of required staff levels might exist that lead to shift schedules with different costs. Two-step staffing algorithms may lead to suboptimal solutions of shift schedules [8, 5].

Henderson and Mason [9] proposed a feed-back approach called iterative cutting plane method which updates required staff levels and shift schedules iteratively. It requires system service quality to be a non-decreased concave function of staffing. Continuing with their work, Atlason et al. [3] relax the concavity assumption to pseudo-concavity. The iterative cutting plane algorithm combines a queueing simulation and an integer programming model to generate required staff levels and shift schedules simultaneously. Queueing simulation tests the feasibility of current staff levels and refines staff level feasible set by adding more linear constraints. The integer programming solution provides simulation input in the next iteration.

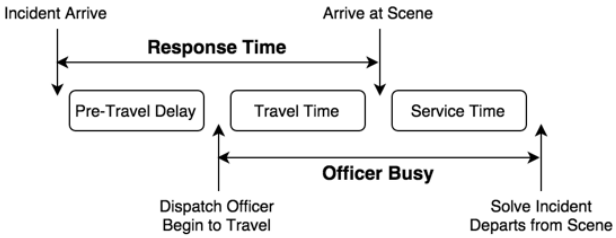

Figure 1 Response process in patrol police service systems.

This algorithm has been widely applied in many other service systems, such as healthcare [1] and delivery service systems [6]. Compared with two-step approaches, some suboptimal solutions for shift scheduling can be avoided in feed-back approaches, but there is an obvious disadvantage that feed-back approaches do not make full use of provided information at the starting point and large computation efforts are required to solve realistic problems.

Some other research has been conducted to construct shift schedules directly without using the concept of required staff level. Ingolfsson et al. [10] use Chapman Kolmogorov equations to evaluate performance of a given schedule. Gans et al. [7] apply integrated forecasting and stochastic programming to generate schedules for call centre staffing. An obvious advantage in applying a direct approach is that they skip the staffing requirement evaluations and focus on the final solution of shift schedules. It will largely reduce the feasible space and make it easier to explore the optimal solution.

In this paper, a new staffing algorithm is proposed to determine appropriate staffing level to fit shift schedules in PPSSs. This algorithm starts at a refined SRS algorithm. This refined square root indicates an appropriate staffing in each period and the corresponding shift schedule. Simulation will then be applied to improve the shift schedule towards the optimal solution.

The remainder of the paper is arranged as follows. Section 2 formally defines PPSSs. A mathematical model for the staffing problem is described in Section 3. A new staffing model based on SRS algorithm is proposed in Section 4 for solving patrol officer staffing problems. Section 5 introduces a simulation combined local search method to modify feasible shift schedules towards the optimal. An example case for PPSS problem is discussed in Section 6. The last section gives conclusions and identifies promising directions for further research.

2 Police patrol service system

Call handlers in contact management departments of police forces are often the first point of contact for the general public. In Leicestershire Police Force, call handlers deal with approximately 16,000 emergency calls and 750,000 non-emergency calls every year [15]. About half of these calls require immediate patrol officer attendance. Depending on reported information of an incident, call handlers will assess the situation and decide the urgent grade this incident should be. Among all the incidents requiring immediate patrol officer attendance, 70% are emergency incidents and the rest 30% are priority incidents. After the grading, free patrol officer will be informed for incident attendance. If no patrol officer is currently available, incidents will wait for service and pre-travel delays will be generated. The response process in PPSSs is illustrated in Figure 1.

When an incident is assigned to an available patrol officer, the patrol officer is labelled as busy until the assigned incident is solved. Response times are recorded as lags between

the time incidents being recorded and the time patrol officers arrive at the scene. Incident arrival rate, incident service time and time spent on travel are predictable with the help of historical records. Identical to queueing systems, a patrol police response system consists of three main procedures: arrival, dispatch and service. The arrival of an incident is based on the time at which it is recorded. The service for each incident starts at the time that assigned patrol officers begin to travel to the incident and will last as long as it takes for the incident to be solved. Dispatch refers to the assignment of patrol officers to attend incidents.

PPSSs are similar to non-preemptive priority queues. Emergency incidents and priority incidents have separate logical waiting queues. When a patrol officer becomes free, the incident at the head of the emergency queue will be the next to be served unless all the waiting incidents are priority. The service of a priority incident will not be disrupted even if emergency incidents arrive. From previous experience in patrol response system operations, short delays in dispatching priority incidents are allowed to exist but emergency incidents always expect immediate responses because of travel times. The difference in target response times of emergency incidents and priority incidents is 45 minutes, this being the allowed short delay in responding to priority incidents. When a 24-hour working day is divided into thirty-two 45-minute periods, a modification can be made to tailor time-dependent staff levels into shift patterns by leaving some priority responses to the next period.

3 General staffing model

Integer programming, as a widely used model in workforce management provides an effective way to find a combination of shift staffing which minimizes workforce cost, subject to the target system performance. It is assumed that the whole planning horizon, usually one day, is divided into I non-overlapping periods. In each period $i \in I$, the service system is evaluated to make sure that it maintains a stable performance. Assume there are J predefined shifts. Matrix A records the relationships between shifts and periods.

$$A_{ij} = \begin{cases} 1, & \text{if period } i \text{ and the time of shift } j \text{ overlaps.} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Let c_j be the cost of working on shift j and x_j are the number of staff who work on shift j. The objective function is

$$\min \sum_j c_j x_j \ . \tag{2}$$

Besides x_j, another set of decision variables y_i are introduced to denote the lower bound of required staff in period i. A feasible combination of x_j should provide at least y_i working staff in period i, so

$$\sum_j A_{ij} x_j \geqslant y_i \ , \quad \begin{array}{l} \text{where } x_j \text{ and } y_i \text{ are integers} \\ \text{for all } i \in \{1, 2, \ldots, I\} \text{ and } j \in \{1, 2, \ldots, J\} \ . \end{array} \tag{3}$$

Minimum required staff levels should provide enough services to meet performance targets. In PPSSs, over 80% of priority incidents should be responded to within the targeted time. Emergency incidents require slightly faster services and at least 85% of them should receive successful responses. For ease of notation, set $\vec{y_i}$ as the vector of staff levels from period 1 to period i.

$$\vec{y_i} = (y_1, y_2, \ldots, y_i), \text{ for all period } i \in \{1, 2, \ldots, I\}. \tag{4}$$

Service quality in period i only depends on the value of $\vec{y}_i$. Any staffing beyond period i will not influence the system performance in period i. For a period $i \in \{1, \ldots, I\}$, an implicit functions $g_i^1(\vec{y}_i)$ is introduced to denote the actual successful response percentages for emergency incidents being reported in period i. Similar to $g_i^1(\vec{y}_i)$, function $g_i^2(\vec{y}_i)$ is for priority incidents. Constraints about service targets for emergency incidents and priority incidents can be formulated as follows,

$$g_i^1(\vec{y}_i) \geqslant 85\%; \quad g_i^2(\vec{y}_i) \geqslant 80\%, \quad \text{for all period } i \in \{1, 2, \ldots, I\} \ . \tag{5}$$

As stated before, incident arrival time, travel time and service time are random variables. Successful response percentages $g_i^1(\vec{y}_i)$ and $g_i^2(\vec{y}_i)$ are also random variables when the values of $\vec{y}_i$ are fixed so they can be approximated by their sample average $\hat{g}_i^1(\vec{y}_i)$ and $\hat{g}_i^2(\vec{y}_i)$. Assume simulations are performed K times with the same configuration of having $\vec{y}_i$ staff on shift from period 1 to period i and the successful response percentage of the kth iteration are denoted as $g_{ik}^1(\vec{y}_i)$ and $g_{ik}^2(\vec{y}_i)$. Constraint (5) can be replaced by

$$\hat{g}_i^1(\vec{y}_i) = \frac{1}{K}\sum_{k=1}^{K} g_{ik}^1(\vec{y}_i) \geqslant 85\%; \quad \hat{g}_i^2(\vec{y}_i) = \frac{1}{K}\sum_{k=1}^{K} g_{ik}^2(\vec{y}_i) \geqslant 85\%, \quad \forall i \in \{1, 2, \ldots, I\} \ . \tag{6}$$

Until now a general optimisation model for staffing problems is established to minimise shift costs (2) with respect to coverage constraints (3) and service target constraints (6).

4 Refined square root staffing model

In this section, the proposed approach which combines a refined SRS and simulation will be described. The general idea of this approach is to distribute required workload in each period and generate an initial shift schedule. Indicated by simulation results, the shift schedule can be further improved towards the optimal.

Consider a $s(t)$-server queueing system $M_t/G/s(t)$ with Poisson arrival process (M_t) and general service time (G). Assume incident arrival rate $\lambda(t)$ is a function of time t and incident service time μ is a random variable with a probability density function $f(\mu)$. According to the work of [11], the offered load at time t is $m(t)$ which means at time t there will be $m(t)$ servers busy on average.

$$m(t) \approx \int_0^t \Big\{\lambda(t-\mu)\cdot F^c(\mu)\Big\}d\mu, \text{ where } F^c(\mu) = \int_\mu^\infty f(v)dv \ . \tag{7}$$

An intuitive explanation for the value of $m(t)$ is as follows. Incidents that arrive at $(t-\mu)$ have the probability $F^c(\mu)$ of remaining in service at time t. Integrating incidents by their arrival time, the value of $m(t)$ is obtained. It is obvious that $m(t)$ provides a lower bound for long time steady state staffing in service systems without any customers who abandon their service requirements while waiting in queues. Allocating $m(t)$ servers at time t is not enough to achieve service targets. Additional servers need to be added in to improve system performances. As a general rule-of-thumb, SRS augments offered load $m(t)$ by an amount of staff that is proportional to the square root of the offered load.

$$s(t) \geqslant \left\lceil m(t) + \beta\sqrt{m(t)} \right\rceil, \tag{8}$$

where β is related to the delay probability $P\{\text{delay}\} = P\{N(0,1) \geqslant \beta\}$.

As stated in [17], SRS algorithm and its variations are remarkably effective for staffing in infinite server service systems and finite server service systems with abandon. In PPSSs,

delayed incidents will keep on waiting even though they have been waiting for a long time. Previous delay will largely effect current system performance. Pre-travel delays in responding to priority incidents is acceptable. As defined in Section 2, a priority incident can be delayed in response by up to one period of 45 minutes. Theoretical analysis for time-dependent multi-grade queue system is too complex to perform, but when aligning target response time of priority incidents to emergency incidents, this multi-grade queue system is reduced to a single-grade queue and the SRS algorithm can be applied in analysing PPSSs. The main idea of the refined SRS is adjusting required staff level to fit shift schedules by setting intended delay in priority incident responses.

Set $t_0 = 0$ as the starting time of simulation, period $i \in \{1, 2, \ldots, I\}$ starts at t_{i-1} and finishes at t_i. Workload $M(i)$ is defined in (9) to represent the integral of offered load $m(t)$ in period i.

$$M(i) = \int_{t_{i-1}}^{t_i} m(v)dv, \quad \text{with } m(v) \text{ defined in (7)} \ . \tag{9}$$

As defined in Section 2, priority incidents in PPSS that account for about 70% of all immediate attendance incidents can wait up to one period of 45 minutes. Set w_i as the delayed workload from period i to period $i+1$ in responding to priority incidents. Assume there are z_i busy servers on average in period i, the provided workload as shown in the right hand side of (10) should be no less than the required workload as shown in the left hand side of (10).

$$M(i) + w_{i-1} - w_i \leqslant z_i \cdot (t_i - t_{i-1}), \quad \text{with } M(i) \text{ defined in (9)} \ . \tag{10}$$

Since only priority responses can be delayed, the workload related to priority incidents in period i sets an upper limit of w_i as shown in (11)

$$w_i \leqslant 70\% \cdot M(i) \ . \tag{11}$$

Constraint (10) and (11) align the service target for emergency incidents and priority incidents in response time, but emergency incidents still require slightly higher successful response percentage than priority incidents. Set the emergency incident target 85% as the common percentage for both grades of incidents, the successful response target for priority incidents 80% will be met but required staff levels may be slightly over estimated. Simulation modifications will help correct these over estimations. Based on the SRS algorithm presented in (8), for the number of average busy server z_i and the required staff level y_i, constraint (12) holds in period i

$$y_i \geqslant \lceil z_i + 1.03\sqrt{z_i} \rceil \ , \quad \text{where } P\{\text{delay}\} = 15\% = P\{N(0,1) \geqslant 1.03\} \ . \tag{12}$$

Please note that (12) is not a linear constraint because of square root part of z_i, but when z_i takes value from interval $[z_i^{\text{lower}}, z_i^{\text{upper}}]$, it can be approximated via the following linear constraint.

$$y_i \geqslant \left\lceil f(z_i^{\text{lower}}) + \frac{f(z_i^{\text{upper}}) - f(z_i^{\text{lower}})}{z_i^{\text{upper}} - z_i^{\text{lower}}}(z_i - z_i^{\text{lower}}) \right\rceil , \quad \text{where } f(z) = z + 1.03\sqrt{z} \ . \tag{13}$$

One way to achieve a reasonable lower bound z_i^{lower} is as follows. Assume in period i there is no work passed over from the previous period. Only emergency incidents that cannot be delayed in response will be attended in this period. It is the best lower bound achievable in terms of workload. Similarly, z_i^{upper} can be set as an upper bound when all workload that

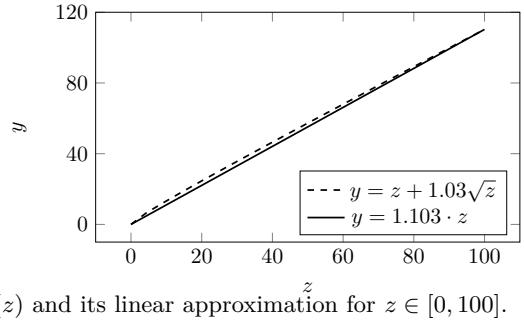

Figure 2 Function $f(z)$ and its linear approximation for $z \in [0, 100]$.

comes from priority responses initiated in the previous period is delayed to period i and there is no work carry over from period i to the next period.

Figure 2 plots function $f(z)$ in green and its linear approximation in blue for $z \in [0, 100]$. It indicates that constraint (13) provides a good linear approximation for constraint (12). The smaller the gap between z_i^{upper} and z_i^{lower}, the closer the approximation is to $f(z)$. Although y_i that satisfies (13) may still violate (12), a second-order cone programming [2] is simplified to a linear programming with these linear approximations. Instead of applying simulations to check the feasibility for service target constraint (6), based on the refined SRS algorithm, a combination of linear constraint (10), (11) and (13) are introduced to define feasible staff level y_i in period i. An integer programming model can be proposed to minimise the cost for shift schedules via objective function (2). Constraint (3) remains the same denoting the relationship between staff level y_i for period i and shift schedule x_j for shift j. This proposed linear programming model can be used to generate an initial shifts schedule which is close to the optimal shift schedule.

5 Simulation combined staffing modification

Simulation of queueing systems will then be implemented to indicate modifications in shift schedules. According to the response sequence in local police force, PPSSs are simulated by static priority queues in which emergency incidents can jump ahead of all priority incidents in the waiting queue but services for priority incidents will not be interrupted by new arrivals of emergency incidents. Simulator in this project is written in C++ using the standard procedure for discrete-event systems as suggested in [12], but it can be easily modelled by other simulation software as well.

In PPSSs, demand for service that comes from emergency incidents and priority incidents depends on an overall time-varying arrival rate $\lambda(t)$. Service process in queueing systems will start immediately when a patrol officer is available upon an incident arrival; otherwise the incident will join the queue. One should note that a service process in PPSSs actually starts at the time when an available patrol officer begins to travel to the assigned incident scene. That is to say the time one patrol officer spends on dealing with one incident also include the time patrol officers spend on travelling. Assume the time one patrol officer spends on one incident lasts long $\mu = \mu_1 + \mu_2$, where μ_1 denotes the time spend on travelling and μ_2 is the time to solve the incident at the scene. In a PPSS with time-varying staff level, the overall service rate $s(t)/\mu$ can be influenced by changing $s(t)$, the number of working patrol officers at time t. Figure 3 displays a simulation model for PPSSs.

Since patrol officers work in shift pattern, a shift schedule $\{x_1, x_2, \ldots, x_J\}$ will decide the number of working patrol officers $s(t)$ at time t. The shift schedule obtained from refined

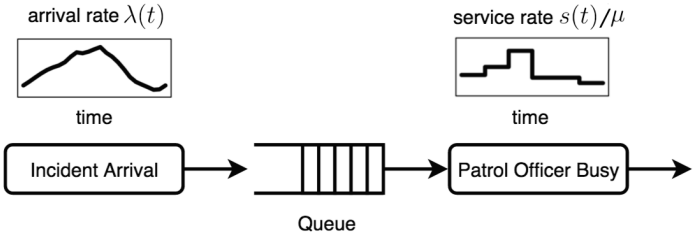

Figure 3 Police patrol service system simulation model.

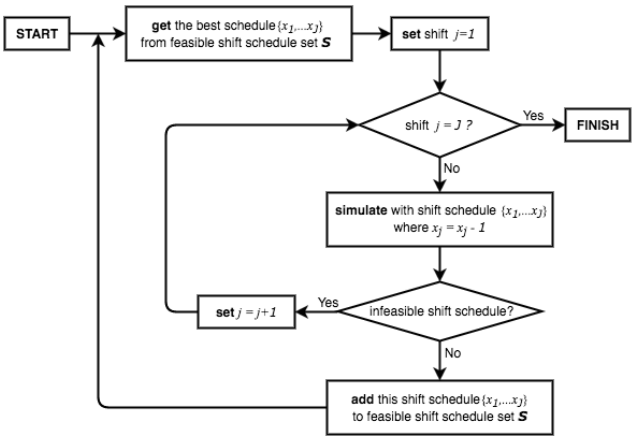

Figure 4 Simulation combined local search for staffing modification.

SRS model will be the first shift schedule operated in PPSS simulation model. For each period $i \in \{1, 2, \ldots, I\}$, both emergency incidents successful response percentage and priority incidents successful response percentage are compared with service targets. If the simulation system fails to meet service targets at period i, additional patrol officers will be arranged to work on related shifts; otherwise, less scheduled patrol officers may still be able to cope with the demand for service. The simulation combined staffing modification uses a local search algorithm as shown in Figure 4.

It starts at the initial shift schedule obtained from refined SRS model as stated in Section 4. Simulation is implemented to check whether the initial shift schedule can deliver qualified service. If service target is violated in period i, a new shift schedule will be generated to arrange one more patrol officer to work in period i. Local search modifications in staff level will continue until there is a feasible shift schedule that meets police patrol service targets at all time. Then local search are then applied to explore potential subtractions in the current feasible shift levels. Besides the easy implementation of local search, a linear objective function of shift levels is another main reason for applying local search. It is unlikely to get trapped in a local optimal shift schedule. Compared with other existing staffing algorithms, such as staffing via exhaustive simulation and iterative feed-back staffing, this local search integrated with refined SRS leads to a large computational saving.

6 Computational study

An example case for patrol staffing in Leicestershire police force is discussed in this section. According to the real patrol activity records in November 2010, input parameters for this simulation model are estimated as listed in Table 1 and Figure 5.

Table 1 Input parameters for police patrol service system simulation.

Urgency Grade	Arrival Rate *Poisson Process*	Service Time *Exponential Distribution*	Travel Time *Normal Distribution*
1. Emergency	30%$\lambda(t)$/hour	1 hour *on average*	5 minutes *on average*
2. Priority	70%$\lambda(t)$/hour		

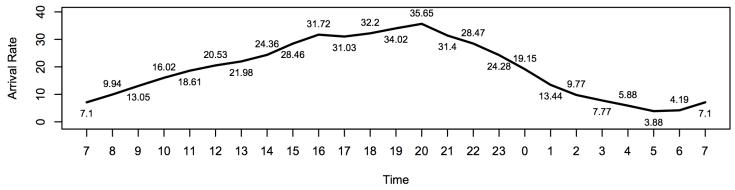

Figure 5 Incident arrival rate $\lambda(t)$ for police patrol service system simulation.

A working day in PPSSs starts at 7am and finishes at 7am the following day. Although PPSSs run continuously from one day to the next, the demand for service around 7am in the morning is very low and the system around that time is almost empty. For this reason, each working day can be viewed as a separated replication. Service levels for emergency incidents and priority incidents in all periods are estimated by performing multiple (1,000) independent replications of simulation. Every working day is divided into thirty-two 45-minute periods and service targets for emergency responses and priority responses will be estimated in all periods. Four different shifts can be performed by patrol officers. They are 'Early' shift from 07:00 to 15:00; 'Day' shift from 15:00 to 23:00; 'Late' shift from 19:00 to 03:00 and 'Night' shift from 23:00 to 07:00. The cost to perform an 'Early' shift, a 'Day' shift, a 'Late' shift or a 'Night' shift is set to £112, £112, £144 and £160, respectively. It considers related factors including hourly pay to patrol officer, vehicle fuel cost and shift type popularity.

In PPSSs, the time a patrol officer spent on an incident μ consists of the time for travelling μ_1 and the time spent on solving incidents μ_2. These two parts of service time come from two different distributions, normal distribution and exponential distribution. It is not easy to get an analytical form of the probability density function $f(\mu)$, but it can be estimated through random variables. Generate a number (10,000) of random variables for travel time μ_1 and generate the same number of random variables for service time μ_2. The distribution of μ can be estimated by random variable pairs of μ_1 and μ_2. Figure 6 plots the approximated function of $m(t)$ for a working day as defined in (7).

The initial shift schedule is generated by optimising the refined SRS model. Simulation combined with local search modifies the initial shift schedule towards the optimal. The initial and the optimal shift schedules for this example are summarised in Table 2.

In this example, the initial shift schedule generated by the refined SRS model is close to the optimal shift schedule, but it slightly over estimates the required staff level during rush hours around 19:00 to 23:00. The over estimation in staff level is mainly because PPSS does not strictly require immediate response without any pre-travel delay to achieve the service targets. With the help of the refined SRS model, the response target for all incidents is aligned so that more than 85% of incidents should be responded within 15 minutes. In this example case, travel times are assumed to follow a normal distribution with mean of 5 minutes, so the time spent on travelling for 85% incidents is less than 6 minutes. A pre-travel delay of up to 9 minutes is permitted. This over estimation can be diminished by tuning the value of safety parameter β in (8), but it is hard to be avoided analytically.

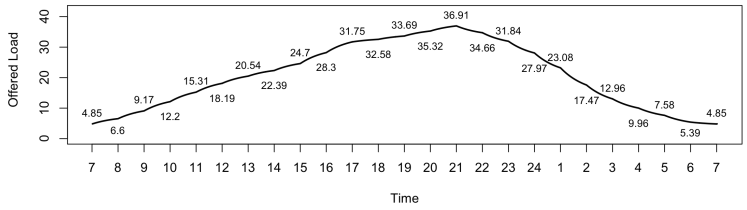

Figure 6 Offered load $m(t)$ for police patrol service system simulation.

Table 2 Initial shift schedule generated by refined square root staffing model.

	Early shift	Day shift	Late shift	Night shift
Initial Shift Schedule	19 officers	29 officers	11 officers	8 officers
Optimal Shift Schedule	19 officers	27 officers	9 officers	8 officers

7 Conclusion and future work

In this paper, a new algorithm is proposed to determine appropriate staffing levels in PPSSs. It starts at a refined SRS algorithm to indicate an initial shift schedule and then uses simulation combined with local search to modify the shift schedule towards the optimal. The proposed approach is believed to be more efficient comparing with other existing staffing algorithms. It is a hybrid method of two-step staffing and feed-back staffing. A good initial shift schedule generated will significantly reduce the computation time of simulation. It also skips the estimation of staff levels by modifying shift levels directly which may further reduce the computation time required to explore the optimal shift schedule.

The proposed staffing algorithm can also be revised to solve other staffing problems. The planning horizon can be extended from a single-day period to a multi-day period when some work initiated at the end of a day would be taken over by the staff scheduled to work at next day. The longer the planning horizon, the more the decision variables and the longer it will take to find the optimal solution via simulation only. Start the simulation with an informed initial shift schedule, as suggested in our approach, will largely reduce the computation time by skipping simulations of some unnecessary scenarios. The current model aligns the priority incident service target to the emergency incident service target and assumes that there is no pre-travel delay in responding to emergency incidents. However, the proposed staffing model is very sensitive to the time patrol officers spend on travelling, a potential improvement in the proposed staffing algorithm could be made by introducing reasonable pre-travel delay in emergency incidents as well.

References

1 Bernardetta Addis, Roberto Aringhieri, Giuliana Carello, Andrea Grosso, and Francesco Maffioli. Workforce management based on forecasted demand. In *Advanced Decision Making Methods Applied to Health Care*, pages 1–11. Springer, 2012.

2 Farid Alizadeh and Donald Goldfarb. Second-order cone programming. *Mathematical programming*, 95(1):3–51, 2003.

3 Júlíus Atlason, Marina A Epelman, and Shane G Henderson. Optimizing call center staffing using simulation and analytic center cutting-plane methods. *Management Science*, 54(2):295–309, 2008.

4 Suffolk Constabulary. Safety advice: police response policy. `http://www.suffolk.police.uk/safetyadvice/reportacrime/policeresponsepolicy.aspx`. Accessed: 2016-05-02.

5 Mieke Defraeye and Inneke Van Nieuwenhuyse. Staffing and scheduling under nonstationary demand for service: A literature review. *Omega*, 58:4–25, 2016.

6 Yixin Diao and Aliza Heching. Staffing optimization in complex service delivery systems. In *Network and Service Management (CNSM), 2011 7th International Conference on*, pages 1–9. IEEE, 2011.

7 Noah Gans, Haipeng Shen, Yong-Pin Zhou, K Korolev, A McCord, and H Ristock. Parametric stochastic programming models for call-center workforce scheduling. *Technical report*, 2009.

8 Linda V Green, Peter J Kolesar, and João Soares. Improving the sipp approach for staffing service systems that have cyclic demands. *Operations Research*, 49(4):549–564, 2001.

9 Shane G Henderson and Andrew J Mason. Rostering by iterating integer programming and simulation. In *Proceedings of the 30th conference on Winter simulation*, pages 677–684. IEEE Computer Society Press, 1998.

10 Armann Ingolfsson, Md Amanul Haque, and Alex Umnikov. Accounting for time-varying queueing effects in workforce scheduling. *European Journal of Operational Research*, 139(3):585–597, 2002.

11 Otis B Jennings, Avishai Mandelbaum, William A Massey, and Ward Whitt. Server staffing to meet time-varying demand. *Management Science*, 42(10):1383–1394, 1996.

12 W David Kelton and Averill M Law. *Simulation modeling and analysis*. McGraw Hill Boston, 2000.

13 The Secretary of State for Transport. The road vehicles (construction and use) regulations. `http://www.legislation.gov.uk/uksi/1986/1078/made`. Accessed: 2016-05-02.

14 The Secretary of State for Transport. The road vehicles lighting regulations. `http://www.legislation.gov.uk/uksi/1989/1796/made`. Accessed: 2016-05-02.

15 Leicestershire Police. Call taker: responding to your calls 24 hours a day. `https://leics.police.uk/join-us/police-staff/call-taker`. Accessed: 2016-05-02.

16 West Midlands Police. Contact us: responding to your calls. `https://www.west-midlands.police.uk/contact-us/response-times/index.aspx`. Accessed: 2016-05-02.

17 Ward Whitt. What you should know about queueing models to set staffing requirements in service systems. *Naval Research Logistics (NRL)*, 54(5):476–484, 2007.

A Column Generation Approach for Pure Parsimony Haplotyping

Veronica Dal Sasso[1], Luigi De Giovanni[2], and Martine Labbé[3]

1 **Dipartimento di Matematica, Università degli Studi di Padova, via Trieste 63, 35121 Padova, Italy**
`dalsasso@math.unipd.it`

2 **Dipartimento di Matematica, Università degli Studi di Padova, via Trieste 63, 35121 Padova, Italy**
`luigi@math.unipd.it`

3 **GOM, Université Libre de Bruxelles, Bd du Triomphe CP210/01, 1050 Bruxelles, Belgium**
`http://gom.ulb.ac.be/`

Abstract

The knowledge of nucleotides chains that compose the double DNA chain of an individual has a relevant role in detecting diseases and studying populations. However, determining experimentally the single nucleotides chains that, paired, form a certain portion of the DNA is expensive and time-consuming. Mathematical programming approaches have been proposed instead, e.g. formulating the Haplotype Inference by Pure Parsimony problem (HIPP). Abstractly, we are given a set of genotypes (strings over a ternary alphabet $\{0, 1, 2\}$) and we want to determine the smallest set of haplotypes (binary strings over the set $\{0, 1\}$) so that each genotype can be "generated" by some pair of haplotypes, meaning that they are compatible with the genotype and can fully explain its structure.

A polynomial-sized Integer Programming model was proposed by Catanzaro, Godi and Labbé (2010), which is highly efficient but hardly scalable to instances with a large number of genotypes. In order to deal with larger instances, we propose a new model involving an exponential number of variables to be solved via column generation, where variables are dynamically introduced into the model by iteratively solving a pricing problem. We compared different ways of solving the pricing problem, based on integer programming, smart enumeration and local search heuristic. The efficiency of the approach is improved by stabilization and by a heuristic to provide a good initial solution. Results show that, with respect to the linear relaxations of both the polynomial and exponential-size models, our approach yields a tighter formulation and outperforms in both efficiency and effectiveness the previous model for instances with a large number of genotypes.

1998 ACM Subject Classification G.1.6 Optimization

Keywords and phrases computational biology, haplotyping, column generation, integer programming, combinatorial optimization

Digital Object Identifier 10.4230/OASIcs.SCOR.2016.5

1 Introduction

One of the most important achievements of the latest years in biology has been the human genome sequencing, completed in 2001, that has shown how all humans share the 99% of the information contained in the DNA, while all the significant differences are contained in the remaining information. Each site of this 1% portion of the human genome,

that presents a significant variability among the individuals, is called a *Single Nucleotide Polymorphism (SNP)*.

Humans are diploid organisms, meaning that the DNA is organized in pairs of chromosomes, each copy coming from one of the two parents. Every single chain in the DNA is made of a sequence of nucleotides, choosing among the possible four: A, T, C, G. It is known that, regarding human beings, the DNA sites and so also the SNP sites are almost always biallelic, meaning that at each site only two of the four nucleotides can be found. If the nucleotide is equal for both chains, then the SNP is *homozygous*, otherwise it is *heterozygous*. From now on, we denote with *haplotype* the single chain of SNP values for a specific portion of a chromosome copy and with *genotype* the chain providing information regarding the union of the two chromosome copies, that tells us if each SNP in the chain is homozygous or heterozygous. Moreover, we say that two haplotypes resolve a certain genotype if, when paired, the information regarding homozygous and heterozygous sites they give is the same provided by that genotype. Haplotypes have an important role in medical and pharmacologic studies, for example to detect diseases or to study the different behaviour of various individuals to the same therapy. Sequencing them is not practical, as it is very expensive and time consuming, while it is easier to experimentally obtain the information stored in genotypes. We are then facing the *haplotyping* problem, that consists in determining the two haplotypes that resolve a given genotype. Several approaches have been used in order to solve this problem, its difficulty consisting in the fact that, once we have k heterozygous SNPs in the same genotype, we have 2^{k-1} possible pairs of haplotypes that can represent it and we need some criteria to chose the right pair. A classical approach is to apply the *Pure Parsimony* criterion, according to which, given a set of genotypes obtained by a family of individuals, we want to select the minimum number of haplotypes that can resolve all the genotypes.

This problem is called the Haplotype Inference by Pure Parsimony (HIPP) problem. It is well known to be NP-hard [13] and different mathematical programming approaches have been investigated. An exponentially large Integer Programming (IP) formulation with an exponential number of variables and constraints is proposed in [9], able to tackle only small size instances. A combinatorial branch-and-bound algorithm is presented in [18], without great improvements on the efficiency. A different model with an exponential number of variables and constraints can be found in [14], based on a set-covering formulation: variables are related to all possible haplotypes and are dynamically generated by a guided enumeration procedure. Other formulations lead to polynomial-sized models, e.g. [4], where the linear relaxation is weak, [3], that presents a three-index formulation, and [5], where families of valid cuts are derived from the formulation in [10] and hybrid models between existing formulations are proposed. The state-of-the-art polynomial size IP formulation is proposed in [6], which is largely efficient on small and medium-size instances. More efficient non-exact approaches to HIPP have been presented, e.g. [17, 11]. This paper investigates an approach for HIPP to be suitable for large instances. Our contribution consists of a new tighter formulation and basic solution algorithms that outperform previous models on some classes of instances.

The remainder of this paper is organized as follows. Section 2 presents the notation and two new formulations: a slight improvement of the model in [6], polynomial in size and based on class representatives, and a new formulation with an exponential number of variables associated to pairs of haplotypes and genotype subsets and polynomial number of constraints. The latter formulation can be solved via a column-generation approach, whose implementation is detailed in Section 3. Finally, we present computational results showing that our approach is suitable for instances with a large number of genotypes.

2 Formulations

The biallelic property of each SNP allows us to describe a haplotype as a sequence of 0 and 1, where each symbol encodes one of the two possible nucleotides for a specific SNP. A genotype instead is represented as a sequence of symbols chosen among the set $\{0, 1, 2\}$ where 0 and 1 indicate homozygous SNPs in which that specific nucleotide is present and 2 denotes a heterozygous site. A genotype g is resolved by two haplotypes h^1 and h^2 if, for each position p, $h^1_p \neq h^2_p$ whenever $g_p = 2$, and $h^1_p = h^2_p = g_p$ otherwise. A haplotype h and a genotype g are compatible if for every homozygous site p of g we have $g_p = h_p$.

In the HIPP, we are given a set of m genotypes with n SNPs, and we want to determine the least-cardinality set of haplotypes that can be used to resolve all the genotypes.

The two formulations we present in the following stem from the fact that, in a feasible solution to HIPP, each haplotype induces a subset of genotypes that are partially resolved by it, and every genotype belongs to exactly two subsets.

The first formulation slightly improves the one in [6], where each genotype subset is indexed according to the first genotype, in a predefined order, belonging to it. As each genotype belongs to two sets, it can happen that the same genotype g_i should identify two different subsets so that a dummy genotype $g_{i'}$ is created and used to identify the second subset. Thus, we dispose of subsets S_i with index i varying in the set $\bar{K} = K \cup K'$, where $K = \{1, 2, \dots, m\}$ and $K' = \{1', 2', \dots, m'\}$. We also define an ordering such that $1 < 1' < 2 < 2' < \cdots < m < m'$. We then introduce a binary variable x_i, $i \in \bar{K}$, that takes value 1 if in the solution there is a haplotype that induces a subset S_i and 0 otherwise. Genotype subsets are described by variables y^k_{ij}, with $i, j \in \bar{K}$ and $k \in K$, taking value 1 if the k-th genotype belongs to subsets S_i and S_j, 0 otherwise. Further binary variables z_{ip}, $i \in \bar{K}$, $p \in P = \{1, 2, \dots, n\}$ records the value of the p-th SNP in haplotype i. Taking into account symmetries and compatibility issues related to genotypes to be explained by the same haplotype, variables y can be defined on a reduced subset T of triplets (k, i, j) [6]. In particular, if g^k is the k-th genotype, we have $T = \{(k, i, j) \in K \times \bar{K} \times \bar{K} \mid (i < j \leq k \wedge j \neq i' \text{ if } k > i) \vee (i = k \wedge j = k')$ and $\forall\, p \in P,\ (g^i_p = g^j_p = g^k_p) \vee (g^k_p = 2 \wedge g^i_p + g^j_p = 1)\}$. HIPP can be formulated as follows.

$$\text{(PIP)} \quad \min \quad \sum_{i \in \bar{K}} x_i \tag{1}$$

$$\text{s.t.} \quad x_{i'} \leq x_i \qquad \forall\, i \in K \tag{2}$$

$$\sum_{(k,i,j) \in T} y^k_{ij} \geq 1 \qquad \forall\, k \in K \tag{3}$$

$$\sum_{(k,i,j) \in T} y^k_{ij} + \sum_{(k,j,i) \in T} y^k_{ji} \leq x_i \qquad \forall\, k \in K, i \in \bar{K} \tag{4}$$

$$z_{ip} \leq 1 - \sum_{(k,i,j) \in T} y^k_{ij} - \sum_{(k,j,i) \in T} y^k_{ji} \qquad \forall\, k \in K, i \in \bar{K}, p \in P : g^k_p = 0, g^i_p \neq 1 \tag{5}$$

$$z_{ip} \geq \sum_{(k,i,j) \in T} y^k_{ij} + \sum_{(k,j,i) \in T} y^k_{ji} \qquad \forall\, k \in K, i \in \bar{K}, p \in P : g^k_p = 1, g^i_p \neq 0 \tag{6}$$

$$z_{ip} \geq y^k_{ij} \qquad \forall\, (k, i, j) \in T, p \in P : g^k_p = 2, g^i_p \neq 0, g^j_p = 0 \tag{7}$$

$$z_{jp} \geq y^k_{ij} \qquad \forall\, (k, i, j) \in T, p \in P : g^k_p = 2, g^i_p = 0, g^j_p \neq 0 \tag{8}$$

$$z_{ip} \leq 1 - y^k_{ij} \qquad \forall\, (k, i, j) \in T, p \in P : g^k_p = 2, g^i_p \neq 1, g^j_p = 1 \tag{9}$$

$$z_{jp} \leq 1 - y^k_{ij} \qquad \forall\, (k, i, j) \in T, p \in P : g^k_p = 2, g^i_p = 1, g^j_p \neq 1 \tag{10}$$

$$z_{ip} + z_{jp} \geq y^k_{ij} \qquad \forall\, (k, i, j) \in T, p \in P : g^k_p = 2, g^i_p = 2, g^j_p = 2 \tag{11}$$

$$z_{ip} + z_{jp} \leq 2 - y^k_{ij} \qquad \forall\, (k, i, j) \in T, p \in P : g^k_p = 2, g^i_p = 2, g^j_p = 2 \tag{12}$$

$$x_i, y^k_{ij}, z_{ip} \in \{0, 1\} \qquad \forall\, i \in \bar{K}, (k, i, j) \in T, p \in P \tag{13}$$

Table 1 Summary of the notation used.

Sets and data		Variables	
K	set of genotypes' indeces	x_i	set with representative g^i is used
K'	set of dummy genotypes' indeces	y_{ij}^k	g^k is in sets represented by g^i and g^j
$\bar{K}$	set of indeces $K \cup K'$	z_{ip}	record the value of SNPs for haplotype i
$\tilde{K}$	set of heterozygous genotypes	λ^q	Q-pair q is used (or not)
Q	set of Q-pairs $q = (h^q, G^q)$	π^k	dual variables associated to (15)
g^k	k-th genotype	μ_p^k	dual variables associated to (16)
h^q	haplotype associated to Q-pair q	χ^k	g^k is in the solution of PP (or not)
G^q	subset of genotypes associated to q	ζ_p	value of p-th SNP in PP solution

Constraints (2) force the dummy genotype to be used only if the real one is already used as a subset's index, and (3) state that each genotype is resolved. Constraints (4) record whenever the haplotype induced by S_i is used. Constraints (5)-(12) guarantee compatibility issues. With respect to the formulation in [6], we eliminated two sets of redundant constraints and we completed the domains of (7), (8), (9) and (10) to correctly set the values of variables $z_{i,p}$ taking into account all the possible cases for the values of g_p^i, g_p^j and g_p^k.

The second formulation we propose uses an exponential number of binary variables λ^q associated to a pair $q = (h^q, G^q)$ made of a haplotype h^q and a subset of G^q, and taking value 1 if the haplotype h^q is used to resolve all the genotypes in G^q. We denote with Q the set of all possible pairs and refer to any of its elements as a Q-pair. The formulation is derived from a compact quadratic IP model based on two-index variables applying a Dantzig-Wolfe decomposition, as detailed in [8].

We notice that if a genotype has no heterozygous SNPs, it is resolved by taking twice a haplotype equal to the genotype, which must be in the solution (fixed haplotypes). We thus focus on the set $\tilde{K} \subseteq K$ of the genotypes with at least a heterozygous SNP and define, for each Q-pair q, a coefficient c_q equal to 0 if h^q is fixed, 1 otherwise, obtaining the following formulation.

$$\text{(EIP)} \quad \min \sum_{q \in Q} c_q \lambda^q + (m - |\tilde{K}|) \tag{14}$$

$$s.t. \sum_{q \in Q:\ g^k \in G^q} \lambda^q = 2 \qquad \forall\, k \in \tilde{K} \tag{15}$$

$$\sum_{q \in Q:\ g^k \in G^q, h_p^q = 1} \lambda^q = 1 \qquad \forall\, k \in \tilde{K},\ p \in P : g_p^k = 2 \tag{16}$$

$$\lambda^q \in \{0, 1\} \qquad \forall\, q \in Q \tag{17}$$

Constraints (15) ensure that each genotype is resolved by two haplotypes, and constraints (16) ensure that for each heterozygous site of the k-th genotype, only one of the haplotypes used to resolve it has a value 1 in that position, forcing in this way the other haplotype to have a value 0 so that the genotype is correctly resolved. A summary of the notation introduced is shown in Table 1.

3 A column generation approach for EIP

Model PIP can be directly implemented and solved by standard IP solvers, whereas EIP has $O(2^{m\,n})$ variables and solving it with standard solvers can be impractical even for small

size instances. A branch-and-price algorithm should be used [2], where a column generation approach is applied to solve its linear relaxation, that we denote as ELP: at each iteration, we solve a *Reduced Master Problem* (RMP) including a subset of variables and, by applying the theorems on duality, we derive a *pricing problem* (PP) whose aim is to provide either a new variable to possibly improve the solution in the next iteration, or a certificate of optimality.

An initial set of variables is needed, to build a first feasible RMP. We use the following heuristic, based on the approach shown in [7]:

1. initialize the set of haplotypes H to the fixed genotypes, if any
2. for each genotype g with at least one heterozygous site:
   a. look for a haplotype $h \in H$ that can be used to resolve g
   b. if it exists, compute the haplotype v such that h and v resolve g and add v to H
   c. otherwise, build h and v from g by respectively assigning values 0 and 1 to the heterozygous SNPs. Add h and v to H.

3.1 Solving the Pricing Problem

Using the simplex method to solve the RMP, a feasible primal solution to ELP is available, together with a dual solution satisfying the complementary slackness conditions: if the latter solution is dual feasible, then both solutions are optimal. The pricing problem aims at finding any violated dual constraint, corresponding to a primal variable with negative reduced cost. By associating dual variables π and μ to, respectively, constraints (15) and (16), and observing that constraints $\lambda^q \leq 1$ are redundant, we obtain the following dual of ELP:

$$\max \sum_{k=\in\tilde{K}} 2\pi^k + \sum_{k\in\tilde{K}} \sum_{p:g_p^k=2} \mu_p^k \tag{18}$$

$$s.t. \sum_{k:\ g^k\in G^q} \pi^k + \sum_{k:\ g^k\in G^q} \sum_{p:\ g_p^k=2, h_p^q=1} \mu_p^k \leq c_q \qquad \forall\, q \in Q \tag{19}$$

$$\pi^k \geq 0 \qquad \forall\, k \in \tilde{K} \tag{20}$$

Let π_{RM}, μ_{RM} be the dual values from the RMP. The pricing problem, with coefficients (π_{RM}, μ_{RM}), can be formulated as:

$$\text{(PP)}\ \max \sum_{k\in\tilde{K}} \pi_{RM}^k \chi^k - \sum_{k\in\tilde{K}} \sum_{p:g_p^k=2} {\mu_{RM}}_p^k \zeta_p \chi^k - c(\zeta) \tag{21}$$

$$s.t.\ \zeta_p \leq 1 - \chi^k \qquad \forall\, k \in \tilde{K}, p \in P:\ g_p^k = 0 \tag{22}$$

$$\zeta_p \geq \chi^k \qquad \forall\, k \in \tilde{K}, p \in P\ :\ g_p^k = 1 \tag{23}$$

$$\zeta_p, \chi^k \in \{0,1\} \qquad \forall\, k \in \tilde{K}, p \in P \tag{24}$$

where variables ζ and χ describe respectively the haplotype and the genotype subset of the Q-pair, the constraints guarantee compatibility, and $c(\zeta)$ is either 0 or 1, depending on ζ configuring a fixed haplotype or not. The PP can be resolved by first considering the fixed haplotypes, one at a time, and then the other haplotypes. In the first case ζ is given, $c(\zeta) = 0$ and PP can be solved by inspection: for each genotype compatible with the fixed haplotype at hand, evaluate $\pi_{RM}^k + \sum_{p:g_p^k=2, h_p=1} {\mu_{RM}}_p^k$ and set $\chi^k = 1$ if it is non negative; then select the haplotype with the largest value for (21). For non-fixed haplotypes, $c(\zeta) = 1$ and PP has a quadratic objective function and can be directly solved using standard solvers, in case after linearizing by means of a two-index variable to represent the product $\zeta_p \chi^k$. We propose here an alternative approach using a smart enumeration of all possible genotype subsets.

▶ **Proposition 3.1.** *The following* Smart Enumeration *procedure solves PP to optimality:*

1. *for each genotype* g^i, $i \in \tilde{K}$, *in a predefined order* $\succ$
   a. *fix* $\chi^i = 1$, *and* $\chi^k = 0$, $\forall\, k : g^i \succ g^k$
   b. *fix* $\zeta_p = g^i_p, \forall\, p \in P : g^i_p \neq 2$
   c. *set* $G(i) = \{g^i\} \cup \{g^j \in G \mid g^j \succ g^i \text{ and } g^j_p + g^i_p \neq 1, \forall\, p \in P\}$
   d. *solve PP restricted to the genotypes in* $G(i)$*; let* α_i *be the corresponding value*
2. *return the solution related to the maximum* α_i

The condition $g^j_p + g^i_p \neq 1$ *in Step 1c ensures that genotypes* g^j *and* g^i *could be resolved by a common haplotype.*

Proof. By fixing variables χ at Step 1a, we obtain a partition of the solution space of PP into $|\tilde{K}|$ subsets. Let $(\bar{h}, \bar{G})$ be a Q-pair associated to a feasible solution within the i-th subset. Since $g^i \in \bar{G}$, $\bar{h}$ has to be compatible with g^i so that fixing variables as in Step 1b does not exclude any feasible solution. Now, let $g^j \in \bar{G}$ with $g^j \neq g^i$. We necessarily have $g^j \succ g^i$, due to the χ-fixing determining the i-th subset. Moreover, $g^j_p + g^i_p \neq 1, \forall\, p \in P$, as otherwise g^j and g^i cannot be resolved by the same $\bar{h}$. Hence no feasible solution in the i-th is lost by fixing ζ (Step 1b) and by restricting to the genotype subset defined in Step 1c, and α_i is the optimal solution of PP in the i-th subset. ◀

Notice that fixing one genotype in the solution allows us to consistently decrease running times, as we can exploit information on homozygous sites to fix some haplotype coordinates and to choose genotypes in a restricted subset.

Before solving PP exactly, a heuristic can be used to quickly find a variable to be added to RMP. We consider a local search algorithm that starts from the fixed haplotype with associated minimum reduced cost (or a random one, if no fixed ones are available) as current solution. Then all the neighbor solutions defined by flipping one coordinate at a time are generated and evaluated. We notice that evaluating a neighbor solution is equivalent to solving PP for a fixed haplotype, which can be efficiently done by inspection. If the best neighbor solution is better than the current one, it is taken as the new current solution, and the procedure iterates, otherwise the procedure stops.

Once a Q-pair $(\bar{h}, \bar{G})$ with negative reduced cost is available, further Q-pairs to conveniently add to RMP can be determined by taking the same haplotype and extending $\bar{G}$ to include further genotypes. The *Extension Procedure* works as follows: we set an ordering on G and add one genotype compatible with $\bar{h}$ at a time to $\bar{G}$; if the reduced cost associated to the variable corresponding to the new Q-pair (formula (21)) is negative, then it can be added to RMP and the procedure iterates, otherwise the procedure stops. Notice that having Q-pairs with largest genotype subsets may improve the convergence of the column generation procedure, since we give the same haplotype the opportunity to resolve more genotypes, so that the objective function is likely to decrease.

3.2 Stabilization

Solutions to model EIP are often highly degenerate, so that it can take several iterations to recognize that the optimal has been reached, since all the variables with negative reduced cost should be added (this is the so-called tailing-off effect). In order to deal with this issue, we derive a lower bound to be used as an alternative termination criterion. To this aim, we

add the following redundant constraint to EIP:

$$\sum_{q \in Q} \lambda^q \leq M \tag{25}$$

where M is a constant large enough to ensure the constraint is always satisfied. In particular, during the column generation procedure we set its value as tight as possible, updating it at each iteration to the current value of the objective function (14). Notice that the dual of ELP and the pricing problem PP change: in particular, the objective functions (18) and (21) has to be increased by $M\,\nu_{RM}$ and ν_{RM}, respectively, where ν_{RM} is the dual variable associated to (25).

▶ **Proposition 3.2.** *Let $\rho_{RM} = (\pi_{RM}, \mu_{RM}, \nu_{RM})$ be the dual variables associated to the optimal solution of the current RMP and z_{RM} the corresponding optimal value. Let $v(\rho_{RM})$ be the value of the optimal solution of the current PP (including ν_{RM}). Then, $LB(\rho_{RM}) = z_{RM} + \min\{0, M\,v(\rho_{RM})\}$ is a lower bound to ELP.*

Proof. We will show that the lower bound corresponds to the Lagrangian Relaxation of ELP where constraints (15) and (16) (but not (25)) are relaxed. The corresponding Lagrangian function is:

$$L(\rho_{RM}) = \sum_{k \in \tilde{K}} 2\pi^k_{RM} + \sum_{\substack{k \in \tilde{K} \\ p: g^k_p = 2}} {\mu_{RM}}^k_p + \eta \tag{26}$$

The sum of the first two addends is related to the dual objective function of ELP and is equal to $z_{RM} - M\,\nu_{RM}$. The value η is equal to

$$\begin{aligned} \eta = \min \quad & \sum_{q \in Q} c_q \lambda^q - \sum_{k \in \tilde{K}} \pi^k_{RM} \sum_{q: g^k \in G^q} \lambda^q - \sum_{\substack{k \in \tilde{K} \\ p: g^k_p = 2}} {\mu_{RM}}^k_p \sum_{\substack{q: g^k \in G^q \\ h^q_p = 1}} \lambda^q \\ \text{s.t.} \quad & \sum_{q \in Q} \lambda^q \leq M \\ & \lambda^q \geq 0,\ \forall\, q \in Q. \end{aligned}$$

Let

$$\tilde{q} = \arg\min_{q \in Q}\ c_q - \sum_{k \in \tilde{K}:\ g^k \in G^q} \pi^k_{RM} - \sum_{k \in \tilde{K}:\ g^k \in G^q} \sum_{p: g^k_p = 2} {\mu_{RM}}^k_p, \tag{27}$$

then η is obtained by setting $\lambda^q = 0$ for all $q \neq \tilde{q}$, and $\lambda^{\tilde{q}} = 1$ if the minimum in (27) is negative, 0 otherwise. Note that this minimum value is exactly the opposite value of PP plus ν_{RM}. ◀

Further convergence issues are determined by dual degeneracy, which requires stabilization techniques (see, e.g. [15]) to prevent "oscillations" of the dual variables. The technique we adopt solves the PP on a convex combination between the values $\rho = (\pi, \mu, \nu)$ of the current optimal dual variables and a stability center $\bar{\rho} = (\bar{\pi}, \bar{\mu}, \bar{\nu})$. This approach has the advantage of exploiting the lower bound defined above and yields a stabilized column generation procedure that can be sketched as follows [16]:

1. set parameters $0 < \Delta < 1$, $\tau \geq 0$ and $\epsilon > 0$
2. initialize the RMP, the stability centre $\bar{\rho} = \rho_0$, $LB(\bar{\rho}) = -\infty$
3. solve current RMP, obtaining the optimal value z_{RM} and the dual variables ρ_{RM}

4. set $\rho_{ST} = \Delta\,\rho_{RM} + (1-\Delta)\bar{\rho}$ and let z_{ST} be the value of the dual objective function computed in ρ_{ST}
5. solve PP with coefficients ρ_{ST}, obtaining the optimal value $v(\rho_{ST})$ and the Q-pair s
6. compute the lower bound $LB(\rho_{ST}) = z_{ST} + \min\{0, M\,v(\rho_{ST})\}$
7. if $LB(\rho_{ST}) > LB(\bar{\rho})$, update $\bar{\rho} = \rho_{ST}$ and $LB(\bar{\rho}) = LB(\rho_{ST})$
8. if the reduced cost of s with respect to ρ_{RM} is negative, add λ^s to the RMP,
9. if $[(z_{RM} - LB(\bar{\rho}))/LB(\bar{\rho})] \leq \tau$, then set $\Delta = 1$
10. if $z_{RM} - LB(\bar{\rho}) < \epsilon$ then stop, otherwise iterate from 3

It is proved that this algorithm yields the optimal solution [16]. The property is based on the following lemmas guaranteeing that, when a *misprice* happens, that is we do not find a variable to be added to the RMP even if we are not at the optimum, the algorithm is always able to update the stability centre, so that we do not get stuck in a non-optimal solution. Here we adapt the proof of the two lemmas to our lower bound definition.

▶ **Lemma 3.3.** *Let s be the Q-pair defined at Step 5. If the variable λ^s does not have a negative reduced cost, then $LB(\rho_{ST}) \geq LB(\bar{\rho}) + \Delta(z_{RM} - LB(\bar{\rho}))$.*

Proof. Denote with $f(\rho)$ the value of the objective function of PP where the coefficients are ρ and the variables assume the optimal values found when solving PP with coefficients ρ_{ST}. Let $\bar{z}$ be the value of the dual objective function computed in $\bar{\rho}$. We have

$$\begin{aligned} LB(\rho_{ST}) &= z_{ST} + M\min\{0, v(\rho_{ST})\} \geq z_{ST} + Mf(\rho_{ST}) = \\ &= \Delta z_{RM} + (1-\Delta)\bar{z} + \Delta M f(\rho_{RM}) + (1-\Delta)Mf(\bar{\rho}) = \\ &= \Delta(z_{RM} + Mf(\rho_{RM})) + (1-\Delta)(\bar{z} + Mf(\bar{\rho})) \geq \\ &\geq \Delta z_{RM} + (1-\Delta)LB(\bar{\rho}) \end{aligned}$$

where the last inequality holds because $f(\rho_{RM}) \geq 0$ by hypothesis and $f(\bar{\rho}) \geq v(\bar{\rho})$. ◀

▶ **Lemma 3.4.** *When a misprice happens, the gap $z_{RM} - LB(\bar{\rho})$ is reduced by at least a factor of $1/(1-\Delta)$.*

Proof. The sequence $\{z^k_{RM}\}_k$, where k indexes the iterations of the stabilized column generation procedure, is not increasing. Thus, we have

$$\begin{aligned} z^{k+1}_{RM} - LB(\bar{\rho}^{k+1}) &\leq z^k_{RM} - LB(\bar{\rho})^{k+1} \leq z^k_{RM} - LB(\rho^k_{ST}) \overset{(*)}{\leq} \\ &\leq z^k_{RM} - LB(\bar{\rho}^k) - \Delta(z^k_{RM} - LB(\bar{\rho}^k)) = (1-\Delta)(z^k_{RM} - LB(\bar{\rho}^k)) \end{aligned}$$

where inequality (*) holds for the previous lemma. Hence

$$\frac{z^k_{RM} - LB(\bar{\rho}^k)}{z^{k+1}_{RM} - LB(\bar{\rho}^{k+1})} \geq \frac{1}{1-\Delta}\,.$$ ◀

We proved that, whenever a misprice takes place, the lower bound increases, so that according to the stabilization algorithm we need to update the stability center. Moreover, the lower bound increases by a factor big enough to ensure the convergence of the lower bound to the optimal solution.

4 Computational results

In this section we report the results obtained from the computational experiments carried out on instances both from the literature and generated on purpose. The former are taken from

the class *hapmap* used in [5]: they are real instances derived from biological data related to chromosomes 10 and 21 over all four HapMap (International HapMap Consortium 2004) populations. The number of genotypes involved varies between 5 and 68, while the SNPs are either 30, 50 or 75. The latter are random instances characterized by a large number of genotypes (*manygen*). In particular, we generated instances with 80, 90 and 100 genotypes and 10, 20 and 30 SNPs (4 per class, for a total of 36 instances), where SNP is heterozygous with a probability between 10% and 40%, and homozygous sites have the same probability of being 0 or 1.

Model PIP and the stabilized column generation approach for ELP have been implemented in C++ using the SCIP 3.2 [1] library and IBM CPLEX 12.4 [12] solver, and have been tested on an Intel Pentium Dual Core E2160 1.8 GHz processor with 4 GB RAM. A time limit is set to 7200 CPU-seconds for all the implementations. Different variants of the column generation algorithm have been implemented, depending on how the pricing problem is solved. The first variant (*ELP+QPP*) looks for a variable with negative reduced cost by first solving PP on fixed haplotypes, then running the local search procedure and, finally, by linearizing the quadratic PP on general haplotypes and solving it with the standard solver. Notice that a procedure is only executed if the previous one fails. The second variant (*ELP+QPPm*) is similar to the first one, but the extension procedure is applied to add more than one variable at each iteration. The third variant (*ELP+SM*) is as the first one, but the smart enumeration procedure is used instead of the standard solver to solve PP. Finally, a fourth variant (*ELP+SMm*) is as the second one, but using smart enumeration.

Notice that the proposed procedures are highly dependent on the order in which we consider the genotypes. The initial heuristic can end up with sets of different cardinality according to how compatible genotypes are ordered, as can be shown with a very simple example: given the genotypes $g_1 = \{01101\}, g_2 = \{22212\}, g_3 = \{10211\}$, we obtain a better solution if we consider the genotypes in the order g_1, g_3, g_2 instead of g_1, g_2, g_3. In smart enumeration, the ordering of the genotypes affects the size of the problem to be solved at each iteration (if we consider a genotype with many homozygous sites, we have many fixed coordinates and, as a consequence, less decision variables). As for the extending procedure, the order considered can change the set of variables with negative reduced cost that are added at each iteration. In our implementation, we consider genotypes ordered according to the increasing number of heterozygous SNPs.

For the parameters of the stabilization procedure, after preliminary calibration, we set the stabilization parameter $\Delta = 0.15$, the tolerances $\epsilon = 0.1$ and $\tau = 0.1$. Moreover, we set the initial stability center to $\pi_0 = \mathbf{0}$.

Tables 2 and 3 detail our results regarding the LP relaxations of model PIP, indicated with PLP, and EIP, indicated with ELP, for respectively the instances in the *hapmap* and the *manygen* class. The first column of each table identifies the solution approach. The following columns point out the percentage of instances solved within the time limit, the Gap, computed as $(z_{INT} - z_{LR})/z_{INT}$, between the integer solution z_{INT} and the solution of the linear relaxation z_{LR} (average, maximum and percentage of instances having integer linear relaxation), and running time (average, maximum and minimum). Note that all the results regarding the Gap and the running time are referred only to those instances solved within the time limit.

We can see that formulation EIP is tighter than PIP, as the Gap is significantly reduced and the percentage of instances having integer linear relaxation is clearly higher. However, for *hapmap* instances the computational time for the column generation approach is not competitive. When we increase the number of the genotypes, as for the proposed random

Table 2 Results for the *hapmap* class of instances

	% solved	% LR-Gap			time (s)		
		average	max	%0-Gap	average	min	max
PLP	100.00	8.28	25.00	13.04	16.97	0.01	270.14
ELP+ QPP	58.33	4.34	22.22	57.14	1261.18	0.65	5759.27
ELP + QPPm	62.50	4.05	22.22	62.50	819.58	0.76	3446.52
ELP + SM	75.00	3.67	22.22	61.11	1598.46	0.50	6267.11
ELP+ SMm	70.83	3.89	22.22	58.82	789.75	0.47	3557.48

Table 3 Results for *manygen* instances

	% solved	% LR-Gap			time (s)		
		average	max	%0-Gap	average	min	max
PLP	100.00	3.06	32.58	50.00	1263.76	491.78	2287.69
ELP + QPP	61.11	0.22	2.43	86.36	646.63	12.13	7128.80
ELP + QPPm	61.11	0.22	2.43	86.36	647.64	11.96	5996.86
ELP + SM	100.00	1.86	25.40	63.89	269.52	7.83	1677.40
ELP + SMm	100.00	1.86	25.40	63.89	209.09	7.80	1047.98

instances, we can see that the new approach is not only theoretically but also practically efficient. Note that, due to the reduced number of SNPs considered for the *manygen* class, even having a larger number of genotypes results in instances tractable using the same time limit set for the *hapmap* class. Moreover, it can be easily seen that solving the pricing problem with smart enumeration sensibly improves results in terms of number of solved instances within the time limit (in particular, all the *manygen* instances are solved). The effect of the extension procedure can be seen in a reduction of the running times.

5 Conclusions

In this paper, we presented and compared two different formulations for HIPP. The first model PIP is linear and polynomial in size, and refines one previous model in literature. The second model EIP has an exponential number of variables and a relatively small set of constraints. Standard solvers are used for PIP, whereas a column generation approach has been devised to solve the linear relaxation of EIP and implemented, taking into account stabilization techniques to improve its efficiency. Computational tests on real and random instances show that EIP is a consistently tighter formulation than PIP, since its linear relaxation solves a remarkably higher number of instance to integer optimality, and the optimality gap is more than halved on average. From the efficiency point of view, EIP shows promising results on instances with a large number of genotypes, since solving the liner relaxation is faster than PIP in this case. Future work includes integrating the proposed column generation algorithm in a branch-and-price procedure to solve EIP, and investigating specialized branching strategies for both PIP and EIP.

References

1 T. Achterberg. Scip: solving constraint integer programs. *Math. Program. Comput. 1*, 1:1.41, 2009.

2 C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

3 P. Bertolazzi, A. Godi, M. Labbé, and L. Tininini. Solving haplotyping inference parsimony problem using a new basic polynomial formulation, 2006.

4 D. Brown and I. Harrower. A new integer programming formulation for the pure parsimony problem in haplotype analysis. *Algorithms in Bioinformatics. Springer Berlin Heidelberg*, pages 254–265, 2004.

5 D. Brown and I. Harrower. Integer programming approaches to haplotype inference by pure parsimony. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 3(2):141–154, 2006. doi:10.1109/TCBB.2006.24.

6 D. Catanzaro, A. Godi, and M. Labbé. A class representative model for pure parsimony haplotyping. *INFORMS Journal on Computing 22.2*, pages 195–209, 2010.

7 A. Clark. Inference of haplotypes from pcr-amplified samples of diploid populations. *Mol. Biol.Evol.*, 7:111–122, 1990.

8 V. Dal Sasso, L. De Giovanni, and M. Labbé. Dantzig-wolfe decomposition of a quadratic ip model for haplotyping by pure parsimony. *Technical report,* http://www.math.unipd.it/∼luigi/manuscripts/HIPP/TR-DM-HIPP-DWdecomposition.pdf, 2016.

9 D. Gusfield. Haplotype inference by pure parsimony. *Springer Lecture Notes in Computer Science No.2676*, pages 144–155, 2003.

10 B.V. Halldórson, B. Bafna, N. Edwards, R. Lippert, S. Yooseph, and S. Istrail. A survey of computational methods for SNPs and haplotype inference. *Proc. DIMACS/RECOMB Satellite Workshop*, pages 26–47, 2004.

11 Y. T. Huang, K. M. Chao, and T. Chen. An approximation algorithm for haplotype inference by maximum parsimony. *Journal of Computational Biology, 12*, 10:1261–1274, 2005.

12 IBM. Cplex optimizer, 1994. URL: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.

13 G. Lancia, M.C. Pinotti, and R. Rizzi. Haplotyping populations by pure parsimony: Complexity of exact and approximation algorithms. *INFORMS Journal on computing 16*, 4:348–359, 2004.

14 G. Lancia and P. Serafini. A set-covering approach with column generation for parsimony haplotyping. *INFORMS Journal on Computing 21*, 1:151–166, 2009.

15 M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research, 53*, 6:1007–1023, 2005.

16 A. Pessoa, M. Poggi de Aragao, and R. Rodrigues. Algorithms over arc-indexed formulations for single and parallel machine scheduling problems, 2008. URL: http://www.optimization-online.org/DB_FILE/2008/06/2022.pdf.

17 L. Tininini, P. Bertolazzi, A. Godi, and G. Lancia. Collhaps: a heuristic approach to haplotype inference by parsimony. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(3):511–523, 2010. doi:10.1109/TCBB.2008.130.

18 L. Wang and Y. Xu. Haplotype inference by maximum parsimony. *Bioinformatics 19*, 14:1773–1780, 2003.

Model Validation and Testing in Simulation: a Literature Review

Naoum Tsioptsias[1], Antuela Tako[2], and Stewart Robinson[3]

1 **School of Business and Economics Loughborough University, Loughborough, Leicestershire, LE11 3TU, United Kingdom**
n.tsioptsias@lboro.ac.uk
2 **School of Business and Economics Loughborough University, Loughborough, Leicestershire, LE11 3TU, United Kingdom**
a.takou@lboro.ac.uk
3 **School of Business and Economics Loughborough University, Loughborough, Leicestershire, LE11 3TU, United Kingdom**
s.l.robinson@lboro.ac.uk

Abstract

Model validation is a key activity undertaken during the model development process in simulation. There is a large body of literature on model validation, albeit there exists little convergence in terms of the definitions, types of validity, and tests used. Yet it is not clear what standards should be taken into consideration to avoid developing what could be considered to be invalid or wrong models. In this paper we examine existing literature on model validation with the view to identifying the existing validation approaches and types of tests used to assess model validity. In this review we focus our attention on three domains that usually overlap in methods and techniques: general Operational Research (OR), Modelling & Simulation (M&S) and Computer Science (CS). We analyze each field to identify the aspects of validity considered including the tests used, the validation approach taken, i.e. the suggested level of validity achieved (if this applies) and the reported outcome. The analysis shows that there are common validation practices used in all three fields as well as new ideas that could be adopted in discrete event simulation. Some main points of concurrence include the lack of universal validation, the continuous need for validation, and, the indispensable need for modelers and users to work closely together during the model validation process. This review provides an initial categorization of literature on model validation which can in turn be used as a basis for future work in investigating how and to what extent models are considered sufficiently valid.

1998 ACM Subject Classification I.6.4 Model Validation and Analysis

Keywords and phrases Validation, Simulation, Literature review, Types of validity, Field Comparisons

Digital Object Identifier 10.4230/OASIcs.SCOR.2016.6

1 Introduction

Validation is considered an important activity as part of the model development process [11, 15, 20, 24, 25]. The validation process is undertaken in order to ensure that the model developed is sufficiently accurate for the purpose at hand [23, 24]. The importance of model validity becomes more crucial when undertaking modelling in facilitated modelling workshops with stakeholders in what is called facilitated simulation modelling [25, 31]. If the model is not considered accurate by the client that would mean that it would not be possible to proceed with planned workshop activities. How could the model validation process be carried

out as part of a facilitated modelling process? In order to answer this question this paper explores model validation, definitions, process and types of tests that are involved.

While there is a large body of literature that considers validation, the existing approaches and main concepts used differ significantly [11]. To address this lack of homogeneity, we examine existing literature on the topic of model development and validation focusing on simulation. This paper presents a different categorization of validity yet our analysis corroborates previous studies. More specifically, we expand the differences not only to Operations Research (OR) but also to Modelling & Simulation (M&S) and Computer Science (CS). We identify definitions, approaches and tests utilized in existing literature, as well as the validation levels achieved – if applicable. In view of the theoretical background analyzed, we focus our attention on authors recognized for their fundamental contributions and papers with multiple citations. Additionally, we categorize and report validity types in order to establish a basis for validating models in facilitated modelling.

The aim of this paper is to acquire a better understanding of model validation by exploring three different fields (OR, M&S, and CS) with the view to identifying common practices and areas for improvement of validation in simulation. The work presented here, explores the process and approach taken during validation with the view to understanding how the decision that a model is valid is derived. This in turn can inform how wrong models developed during facilitated workshops could still be used to create learning amongst workshop participants.

This paper is structured as follows: Section 2 presents the existing work on model validation, the criteria employed for selecting papers and the areas of investigation. Section 3 provides the analysis of the literature review in order to define commonalities and differences (definitions, validity types, tests, and other aspects of validation), also demonstrating our categorizations of the concepts involved. Section 4 concludes the study with limitations and future research directions.

2 Existing work and literature selection criteria

There are very few studies where authors have undertaken a literature review analysis on model validation to form a coherent description of the differences and to infer commonalities. Pala et al. [21] compare between Hard OR, Soft OR and System Dynamics by mentioning aspects and functions of validation per case. Another relevant study [11] categorizes tests by creating a hierarchy pattern in order to identify the point at which to cease the validation process using a heuristics approach. Nance and Arthur [18] compare three life-cycle models in simulation and highlight the lack of software requirements engineering employment in M&S studies. A second more relevant study researches social sciences' validation, explaining the many problems that occur but doesn't specify techniques [13].

These studies have approached model validation from different perspectives. We address problems and definitions encountered, along with providing a taxonomy of certain ideas. In some occasions these are not clear and the classifications are made based on the authors' interpretation and terms used, especially for types of validity and analysis.

The papers included in our review are selected based on three different criteria: (i) demonstrate a modelling process/life-cycle including validation, (ii) present tests that can be used in validation and relevant procedures (e.g. [2]), (iii) contain examples or cases where validation was implemented explicitly in the paper (quantitatively or qualitatively). The selection or exclusion of the papers analyzed here was performed based on their content of validation and tests as specified from these criteria. Since we are not approaching a systematic review, but rather focus on the most important authors especially in theoretical explanations,

our aim is to provide a clear understanding of the main concepts suggested thus far. To the best of our knowledge, we have sought to include all of the key sources attributed to model validation. Three fields are investigated in this paper: general Operational Research (OR) – also including management and references to other social sciences, Modelling & Simulation (M&S), and Computer Science (CS). The reason we choose these specific domains is due to their overlap with simulation – as it can be indicated by various studies we utilize in the literature (e.g. [34, 27, 12] for CS and [14, 21] for OR). Our selection is justified initially by the apparent use of M&S techniques in OR, and subsequently by the commonalities existing between CS and M&S. More specifically, Balci [2] mentions important differences that "*exist between simulation software engineering and other types of software engineering*" (an opinion shared by M&S [32] and CS [5] papers) but concludes that they can still be applied sideways. Based on this opinion, we could extrapolate the work of Zelkowitz & Wallace [36] where the authors investigate and categorize literature on technology validation, and consider that CS validation outcomes – although different as they may seem – can still be acknowledged as useful and relevant in OR and M&S.

3 A literature review analysis

In this Section we give the most usual definitions and types of validity found in the literature. Also, we comment on the tests used and other aspects relevant to validation as identified in literature.

3.1 Definitions

The evolution of OR – described by Landry et al. [14] – witnessed many additions to the concept of validation. Other than verification, which is mostly attributed to the programming part of model development, we encounter the ideas of independent verification and validation (IV&V) [29], validation, verification, and testing (VV&T) [2], terms like utility [33, 9], certification [33], representativeness [14], assessment [9, 32], accreditation [4], etc. Unlike OR and M&S, the CS papers examined do not try to distinguish that far into the different facets of validation and relevant concepts – even the ones reviewing literature.

Validation itself is viewed as a process and an evidence for "*building the right model*" [2, 23, 20, 13]. One of the earliest examined works – still tracing OR back to its military roots – emphasizes how validation involves the comparison of model outputs with our knowledge of the real world or system [33]. This idea is repeated in other OR [9, 10] and CS [22, 5, 27, 13] adaptations. Sargent states how a model should produce adequate accuracy for its intended use and experimental pre-conditions [28, 29]. This view is also encountered both in the early 1980s [14, 10, 30] and in more recent papers [32, 11] where a necessary "*degree*" of adequacy is considered in order to achieve fidelity. Another suggestion is the usefulness that validation should bring to the model, along with usability, cost consideration and representativeness [9]. Groesser & Schwaninger [11] add that validation is integrated in the model building process. These different views are similar with part of Pala et al.'s analysis [21] – though they see it from a different point of view and purpose of categorization. We should also notice the absence of clear validation analysis in some papers – even those tackling theoretic grounds (e.g. [19, 30]). The same applies for most articles presenting a case study.

Verification is the next most referenced concept found in life-cycle models. It is concerned with "*building the model right*" [2, 23, 20, 13] or achieving a consistent and debugged code [10, 21, 32], to ensure the model runs as intended [33, 9, 20, 29, 13]. It is also accepted that verification regards the correct transition from the designer's conceptual description

[10, 1, 32, 29] or the mathematical/logical model [33, 9, 32] to a numerical one. Gass references both ideas in different papers [9, 10] and Thacker in the same [32]. Even more importantly, an imminent problem occurs when accounting how in social sciences there is a mixed cross-examination of the notion. This overlapping is pointed out by Hahn [13], where the author equates the terms of verification and internal validity. We find more examples in "*verificational validity*" [19, 2], or white-box used as a validity type rather than a technique [23].

Confidence and credibility, used interchangeably, have been connected with the users' idea of whether the model is considered as having credentials for use [14, 9, 10, 21, 20, 29]. Gass [9] and Pace [20] highlight the subjectivity and bias underlying this notion. Other authors [19, 20, 32] view confidence as model accuracy in quantified results. A minimum confidence threshold is suggested [14], with credibility's initial number set to nil [10].

Accreditation is explained analytically in Gass [10], as a broader idea with a comparable content to validation: specific acceptability criteria for an intended use. This is supported by other works [20, 32, 29]. Accreditation is also considered as a similar activity to certification [10, 32] and it is carried out on existing documentation [10]. Certification is considered to be the written assurance of a model's operational applicability [33, 19, 32].

Documentation, accordingly, receives more attention in earlier papers. The necessity and need for using evaluation criteria for all interested parties is pointed out [33], from the beginning of the process. Gass [9] states that it is the "*sine qua non of model validation and assessment*". Other papers supplement these statements [9, 10, 29]. Based on the introductory explanation of how literature handles validation and relevant terminology, we notice that different words are used in the same context and the same word may have different descriptions. The actual problem though isn't just the lexicographic differences that we find, but rather the confusion that arises for modellers and users.

3.2 Types of validity

We now investigate the different aspects of validity encountered in literature in the three fields. We found forty or so different names for validity types, some of which held an elaborate explanation, others were simply referenced, while the rest appeared in contextual description. We implement the same strategy and present the most common ones used, along with our categorization.

Data validity is most often cited in papers. It involves activities leading to: appropriateness, accuracy, completeness, correctness, impartiality, sufficiency, maintainability, reliability, limited cost, and availability of soft and hard, non-biased and biased data [33, 14, 19, 2, 23, 21, 29]. It becomes apparent how potential modellers would get confused over which features should receive the highest priority. Other data aspects involve its transformation [33]. Both Sargent [29] and Gass [10] reference possible constraining issues to consider. Gass [9] distinguishes between raw and structured data. Essentially, data validation should be occurring throughout any and all stages of model development providing some sort of data usage.

Conceptual validity is also considered as an important validation activity. The definitions found so far can be grouped as follows: those concerned with correctness (these include: credibility, relevance, completeness) and the assumptions of the conceptual model, those referring to sufficiency (e.g. level of details/characteristics, fidelity and scope), and the ones explaining relationships (stated and implied variables, system theories and their reasonableness) [33, 14, 9, 2, 23, 21, 20, 29]. Each problem should be validated with a specific purpose in mind and the outcome compared with the real world [9, 23, 29]. Other than these common

points, we mostly find dissimilar explanations for the concept. For example, Landry et al. [14] suggest the use of multiple conceptual models to address each problem, Oral & Kettani [19] mention that this validity should be concerned with how mental databases (that is a type of database capturing the knowledge of the participants of a modelling process) are gathered and utilized, the US GAO report [33] refers to internal logic check etc.

Operational validity is another set of definitions found in the literature. Sufficiency is again found as a common concept, this time on the quality and correspondence of the model's outputs as opposed to real world's data [14, 9, 30, 23, 21, 20, 29, 13]. The main motive is to provide decision-makers with adequate information to accept or reject the model [14, 19, 30, 21]. It also takes into consideration model usability, usefulness, timeliness, synergism, justified time, effort, and costs of the model [33, 14, 19, 30].

Logical – also used as mathematical – validity refers to the translation of conceptual model, through numerical interpretation [14, 9, 19, 21]. This type's content could be considered as one of the reasons for the confusion between validation and verification, as Hahn [13] points out. Indeed, Landry et al. [14] and Pala et al. [21] attribute verification as part of validating the logical process, with the second paper mentioning how "*Sargent calls this step 'computerized model verification'*".

The final most commonly found aspect is that of experimental validity. Authors refer to this type as the process considering quality, efficiency, sufficient accuracy and robustness of solutions, mechanisms and techniques used [14, 19, 21]. The experimenting phase of model development bridges any inconsistencies between the formal model acquired from verification and the final model to be presented in the operational part. We should note once more that "*the level of insight gained*" mentioned by Oral & Kettani [19] is another indication for the need to quantify fidelity in validation.

Having explained the most usual types of validity found, we now focus our attention on other types introduced or referenced by authors. Structural validity is identified by various works [9, 4, 21] as the adequate matching in structure between model and system behavior. Not far away from that lies behavioral validity, acknowledged in the testing section [4, 21, 11] for pattern and structural examination. The idea of theoretical or theory validity is given by the USA GAO report [33] and Sargent [29] and is also mentioned in Smith [30] as bridging the real and modeled world through consistency in theories and assumptions. Gass [9] makes references to technical, model, and dynamic aspects, while Smith [30] apposes comfort, pragmatic, convergent, axiomatic, and criterion validity. We find the use of "*formulational*" and the more confusing "*verificational*" types in Oral & Kettani [19] and Balci [2]. Another set of validation types are mentioned in Hahn [13], including some similar adaptations with terminology such as "*translation*", "*content*", "*discriminate*" etc. Experimental design as part of validation seems to be a case relevant to experimental validity [2, 16, 32]. An interesting category is that of "*communicative*' and "*presentation*" VV&T by Balci [2], which we call "*representation validity*". Predictive validity has two different meanings [9, 13] and it can be also considered as a technique [14, 2]. The terms "*white-box*" and "*black-box*" are indicated both as techniques [2] and as validation steps [23]. The same occurs with face validity mentioned as a technique (e.g. [2, 29]) and as a validation type [10, 13].

From the above, it can be concluded that there is a lack of coherence between the different validation activities. We propose the following initial taxonomy for the aspects of validity encountered thus far in literature:

- Conceptual validity (C): Conceptual, Model, Theoretical, Theory, Formulational, and Specification validity
- Logical validity (L): Logical, Mathematical, and Internal validity

- Experimental validity (E): Experimental, Solver, and Experiment design validity
- Operational validity (O): Operational, Implementer, Dynamic, Comfort, Pragmatic, Convergent, Axiomatic, Criterion, Aptness, Results, Implementation, Replication, White & Black box, External, Convergent, Discriminant, Concurrent, and Predictive validity
- Behavioral validity (B): Structural, Behavioral, Legitimational, Verificational, Construct, Translation, Face, and Content validity
- Representation validity (R): Communicative and Presentation validity
- Data validity (D): Data, Technical, and Technological validity

Concepts can and may overlap, yet we choose to distinguish them based on the required clarity. No distinction can be perfect, yet we employ this grouping to divide the dissimilar phases of the development process with the scope of further using it in future works. Indeed, this classification puts together akin ideas, allowing us to proceed with the tests and approaches part.

3.3 Tests for validation and in-paper employment with results

The plethora of existing tests throughout the literature would make it impossible and out of scope to cite and describe each and every one of them. Balci [2] has created an in-depth analysis for a long list of tests presented there. We will instead have a brief introduction to their usage and discuss the results that are attributed to them.

Tests vary from simple guideline methods for conceptual implementation, to techniques on data checking, verification, experimental and other operational implementations for proving a model capable of sufficient prediction. We consider that some of these approaches can be attributed to general methodologies for the construction and validation of models, like linear programming [33], Toulmin Framework [19, 30], System Dynamics [33], IV&V/third party validation [30, 29], Bayesian analysis [32, 5], artificial intelligence, expert systems, genetic algorithms, fuzzy logic, machine learning [20] etc. We find Gass [10] stating certain criteria against which the model should be scored (based on threshold values and intensity level). Also, there are quite a few references to tests for verification (e.g. [32, 29]).

The actual issue occurs when and if papers follow suggestions for which test to use per validity type. Most of the studies that contain life-cycle models and validation types in OR and M&S provide general guidelines rather than factual data to exemplify. Some exceptions with examples exist, but are mostly challenged qualitatively [9, 30]. Barlas [4] and Gass [10] on the other hand provide examples, focusing mainly on accreditation. Case studies are significantly fewer and only employ or mention partially the theoretical dictations of the previous papers. For instance, Athanassopoulos [1] compares empirical results for a fire department, on a qualitative analysis based on Oral & Kettani's tetrahedron [19], while Longaray et al. [17] approach a Soft OR scenario. However, their results are left vague.

In contrast, an analytical explanation is rarely encountered in CS. Numerical examples on frameworks and quantified results are often presented (e.g. [6, 5, 27, 12, 26] from different areas of CS domain), while the level of validation – if any – is just mentioned or simply omitted.

To quote Groesser & Schwaninger [11], "*the existing categorization of the validation tests as well as the validation processes proposed in the literature are often perceived as too abstract and unspecific to be readily applied*". Table 1 presents the main work reviewed in order to acquire a first impression of how studies perceive tests and the level of validation. The columns depict the three fields we have examined. These are compared with regards to: the usual content found (guidelines, examples, and case studies – row "*Contents*"), the frequency

that definitions appear and are explained (row "*Definitions*"), the areas of application of the cases examined (i.e. where do the authors apply their theoretic implementations or their examples – row "*Areas of application*"), examples that concern validation or a similar concept (row "*Examples*"), the level of validation achieved (row "*Level of validation*"), and tests often recurring in literature (row "*Common validation tests*"). Note that on "*Level of validation*", we cite the outcome that papers present on validation in view of their approach – in other words how the authors consider their validation level. It refers to the general validity of a model under consideration, rather than a unique type of validity. We term as "*vague*" guidelines or case studies where validation is not subjected to an actual outcome.

Based on Table 1 we can draw some conclusions on the common features of each area. We find that OR and M&S provide general guidelines rather than using data to infer validity, contrary to CS papers. This is also depicted on many "*vague*" outcomes existing in the first two fields. On the other hand, CS case studies do not expand on definitions and types of validation. We also notice how OR and M&S areas of applications have many commonalities.

The analysis indicates common validation practices used in the three fields. This could also lead to new ideas adopted in simulation, and more specifically discrete event simulation (DES). For example, we could employ the list of available techniques mentioned here to validate each step during the process and the discrete variables of simulation. Additionally, based on the lack of concrete definitions for validation levels and in combination with our proposed validation types, we could create quantifiable levels for the different validity aspects of such a discrete system in order to describe a model as "*sufficiently*" valid or "*wrong*". Finally, the variety of areas implemented especially in CS provides the opportunity for DES to be tested as well in various fields not yet explored.

The most important conclusion is that our hypothesis about the confusion occurring in test selection is found in all of the fields. This is due to the lack of practical cases in order to quantify sufficiently the appropriateness of each approach [34].

3.4 Other aspects identified in the literature

We now refer to some common ideas found in the literature.

A number of papers refer to the need for identifying acceptable ranges of degrees of validation [9, 10, 19, 30, 21, 11, 29] or adequate trust on the model's behavior and usability (i.e. not entirely correct but at least functional models) [33, 30, 23, 32, 11]. Also papers point out how study objectives should drive the validation requirements [14, 2, 34, 5, 26]. Another concept regards the clients' involvement (as for example teams of users, decision or policy makers, managers, stakeholders, etc.) where close interaction and coordination between modeller and users is recommended (e.g. [33, 14, 9, 10, 19, 30, 21, 16, 32, 29, 35]) with Balci [2] proposing the utilization of Decision Support Systems for a non-technical explanation of the outcomes. Also, Groesser & Schwaninger [11] mention that modellers learn from participating in the process. We could comment that although studies spend a lot of effort highlighting the importance of user involvement, they rarely explain or discuss the exact role of model creators or builders. For example, evaluators are mentioned as independent investigators and multidisciplinary teams for model evaluation [33]. Landry et al. [14] state how model assessors are "*a group independent of both model builders and model users*". Oral & Kettani [19] suggest different types of OR analysts. Lastly, we encounter three very important concepts. First, the lack of perfect or unique validation in models [33, 14, 9, 19, 2, 22, 23, 21, 16, 32, 5, 11, 29] leads to a trade-off in validity [1, 11]. This occurs due to the fact that high validation level of one type doesn't necessarily imply high validation level to another [14, 19, 3, 29]. Second, the need for keeping validation iterative

Table 1 Comparison between fields.

	OR	M&S	CS
Content	Mostly guidelines and/or examples	Mostly guidelines and/or examples	Mostly empirical results from case studies
Definitions	Many definitions usually divergent	Many definitions usually divergent	Lack of definitions
Areas of application	Policy analysis and decision making (e.g. public, government, military) [33, 9] Hard OR [21] Soft OR [17, 21] Systems Dynamics [21]	Simulation models (e.g. government, industry, military) [2, 6, 23, 20, 32, 34, 27, 29] Hard OR [21] Soft OR [21] Systems Dynamics [11, 21]	Simulation models [6, 34, 27] Time-domain models [22] Caching computer-memory strategies [6] Knowledge-based systems [5] Software process model [12] Social networks [26]
Examples	Expected Utility model [30] Fire department (public services) [1]	Epidemics [4] Market growth [4]	Reference generator programme [6] Blast furnace control [16] Mechanical network [34] Thermal system [34] Spot weld [5] Hypersonic wind tunnel [27]
Level of validation	Vague [33, 14, 9, 19, 21, 13, 17] Scores for validation (Poor to Superior) [10] Qualitative evaluation of validity [30]	Vague [2, 23, 21, 20, 32, 11, 29] Validation assumed as "*already passed*" by author [4]	Vague [5, 13] Mathematical conditions for validation [22] Analytical model is robust [6] Model completed and validated [16] Various outcomes – good and bad [34] Metric comparison [12, 26]
Common validation tests	Comparison with (existing) data [33, 14, 9, 21, 13] Statistical tests [33, 14, 9, 21] Face validity [33, 14, 9, 10, 21] Turing test [14] Graphics/Animation [21] Qualitative analysis [30]	Comparison with (existing) data [14, 23, 21, 20, 34, 29] Statistical tests [14, 4, 23, 21, 20, 27, 29] Face validity [14, 21, 20, 34, 29] Turing test [14, 23, 11, 29] Graphics/Animation [4, 21, 29] Qualitative analysis [4, 20]	Comparison with (existing) data [6, 34, 13] Statistical tests [34, 27] Face validity [34]

We consider papers [14] and [21] having references on both OR and M&S.
We consider papers [6], [34] and [27] having references on both M&S and CS.
We consider paper [13] having references on both OR and CS.
Balci [2] is not referenced in tests.
Simulation and general methodologies cited earlier are excluded from tests.

throughout all of the modeling process is often mentioned [33, 14, 19, 2, 3, 8, 23, 32, 5, 11, 29]. And finally, validation is considered to be inseparable from the modelling process [14, 2, 11].

4 Conclusion

This paper explored the existing literature on model validation. Papers from three scientific domains – OR, M&S, and CS – that have commonalities with simulation have been included in the analysis.

The main aspects considered were: definition of validation, the types of validation, the tests used, along with their perspective towards outcomes and suggested level of validity. The most imminent concerns that arose included the use of insufficient terminology for definitions, different types of validity, disparate use of types, confusion of validity type names with techniques, questions on which validity tests should be employed per case, the failure to establish a proper linkage between theory and practice, and the lack of empirical studies with results being vague and the level of validity undetermined.

Needless to say, this initial work is still subjected to limitations. The existing literature especially in the field of CS can be further explored. Also, more tests can be identified to create a more extensive list of techniques and matching aspects. Additional case studies are also required in OR and M&S to verify our hypothesis. The analysis indicates the common validation practices used in the three fields as well as new ideas that could be adopted in simulation, and more specifically DES. This study presents the preliminary analysis of our results. Brooks & Tobias [8] suggest using a set of criteria for selecting a "*correct*" model, contrary to Box & Draper who start from the hypothesis that "*all models are wrong*" and question "*how wrong do they have to be to not be useful*" [7]. Also, Groesser & Schwaninger [11] ask "*how can we transfer to the client the knowledge about the relationship between model structure and the behavior it produces*". Based on this, we will next look facilitated simulation modelling and the definition of models' level of validity in order to define (i) model fidelity and (ii) model sufficiency in providing useful information to clients.

References

1 A. D. Athanassopoulos. Decision Support for Target-Based Resource Allocation of Public Services in Multiunit and Multilevel Systems. *Management Science*, 44(2):173–187, 1998. doi:10.1287/mnsc.44.2.173.

2 O. Balci. Validation, verification, and testing techniques throughout the life cycle of a simulation study. *Annals of Operations Research*, 53:121–173, 1994. doi:10.1109/WSC.1994.717129.

3 O. Balci. Principles and techniques of simulation validation, verification, and testing. In *Winter Simulation Conference Proceedings, 1995.*, pages 147–154, 1995. doi:10.1109/WSC.1995.478717.

4 Y. Barlas. Multiple tests for validation of system dynamics type of simulation models. *European Journal of Operational Research*, 42(1):59–87, 1989. doi:10.1016/0377-2217(89)90059-3.

5 M. J. Bayarri, J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin, and J. Tu. A Framework for Validation of Computer Models. *Technometrics*, 49(2):138–154, 2007. doi:10.1198/004017007000000092.

6 A. J. Bennett, T. Field, and P. Harrison. Modelling and validation of shared memory coherency protocols. *Performance evaluation*, 27&28:541–563, 1996.

7 G. E. P. Box and N. R. Draper. *Empirical model-building and response surfaces*. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley, 1987.

8 R. J. Brooks and A. M. Tobias. Choosing the best model: Level of detail, complexity, and model performance. *Mathematical and Computer Modelling*, 24(4):1–14, 1996. doi:10.1016/0895-7177(96)00103-3.

9 S. I. Gass. Decision-Aiding Models: Validation, Assessment, and Related Issues for Policy Analysis. *Operations Research*, 31(4):603–631, 1983. doi:10.1287/opre.31.4.603.

10 S. I. Gass. Model accreditation: A rationale and process for determining a numerical rating. *European Journal of Operational Research*, 66(2):250–258, 1993. doi:10.1016/0377-2217(93)90316-F.

11 S. Groesser and M. Schwaninger. Contributions to model validation: hierarchy, process, and cessation. *System Dynamics Review*, 28(2):157–181, 2012. doi:10.1002/sdr.

12 H. Gull, S. Alrashed, and S. Z. Iqbal. Validation of Usability Driven Web based Software Process Model using Simulation. *Procedia Computer Science*, 62:487–494, 2015. doi:10.1016/j.procs.2015.08.520.

13 H. A. Hahn. The conundrum of verification and validation of social science-based models. *Procedia Computer Science*, 16:878–887, 2013. doi:10.1016/j.procs.2013.01.092.

14 M. Landry, J.-L. Malouin, and M. Oral. Model validation in operations research. *European Journal of Operational Research*, 14(3):207–220, 1983. doi:10.1016/0377-2217(83)90257-6.

15 A. M. Law and D. M. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill Higher Education, 3rd edition, 1999.

16 M. Le Goc, C. Frydman, and L. Torres. Verification and validation of the SACHEM conceptual model. *International Journal of Human-Computer Studies*, 56(2):199–223, 2002. doi:10.1006/ijhc.2001.0521.

17 A. A. Longaray, L. Ensslin, S. R. Ensslin, and I. O. da Rosa. Assessment of a Brazilian public hospital's performance for management purposes: A soft operations research case in action. *Operations Research for Health Care*, 5:28–48, 2015. doi:10.1016/j.orhc.2015.05.001.

18 R. E. Nance and J. D. Arthur. Software Requirements Engineering: Exploring the Role in Simulation Model Development. In *Proceedings of the Third Operational Research Society Simulation Workshop (SW06)*, pages 117–127, 2006.

19 M. Oral and O. Kettani. The facets of the modeling and validation process in operations research. *European Journal of Operational Research*, 66(2):216–234, 1993. doi:10.1016/0377-2217(93)90314-D.

20 D. K. Pace. Modeling and simulation verification and validation challenges. *Johns Hopkins APL Technical Digest (Applied Physics Laboratory)*, 25(2):163–172, 2004.

21 Ö. Pala, J. A. M. Vennix, and J. P. C. Kleijnen. Validation in Soft OR, Hard OR and System Dynamics: A Critical Comparison and Contribution to the Debate. In *The 17th International Conference of The System Dynamics Society and the 5th Australian & New Zealand Systems Conference*, 1999. URL: http://www.systemdynamics.org/conferences/1999/PAPERS/PARA199.PDF.

22 K. Poolla, P. Khargonekar, A. Tikku, J. Krause, and K. Nagpal. A time-domain approach to model validation. *Transactions on Automatic Control*, 39(5):951–959, 1994. doi:10.1109/9.284871.

23 S. Robinson. Simulation model verification and validation: Increasing the users' confidence. In *Proceedings of the 1997 Winter Simulation Conference*, pages 53–59, 1997. doi:10.1145/268437.268460.

24 S. Robinson. *Simulation: The Practice of Model Development and Use*. Palgrave Macmillan, 2nd edition, 2014.

25 S. Robinson, C. Worthington, N. Burgess, and Z. J. Radnor. Facilitated modelling with discrete-event simulation: Reality or myth? *European Journal of Operational Research*, 234(1):231–240, 2014. doi:10.1016/j.ejor.2012.12.024.

26 N. Ronald, T. Arentze, and H. Timmermans. Towards process validation for complex transport models: A sensitivity analysis of a social network-enhanced activity-travel model. *Computers, Environment and Urban Systems*, 55:24–32, 2016. doi:10.1016/j.compenvurbsys.2015.09.005.

27 C. J. Roy and W. L. Oberkampf. A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing. *Computer Methods in Applied Mechanics and Engineering*, 200(25-28):2131–2144, 2011. doi:10.1016/j.cma.2011.03.016.

28 R. G. Sargent. Some approaches and paradigms for verifying and validating simulation models. In *Proceedings of the 2001 Winter Simulation Conference*, pages 106–114, 2001. doi:10.1109/WSC.2001.977367.

29 R. G Sargent. Verification and validation of simulation models. *Journal of Simulation*, 7(1):12–24, 2012. arXiv:10, doi:10.1057/jos.2012.20.

30 J. H. Smith. Modeling muddles: Validation beyond the numbers. *European Journal of Operational Research*, 66(2):235–249, 1993. doi:10.1016/0377-2217(93)90315-E.

31 A. A. Tako and K. Kotiadis. PartiSim: A multi-methodology framework to support facilitated simulation modelling in healthcare. *European Journal of Operational Research*, 244(2):555–564, 2015. doi:10.1016/j.ejor.2015.01.046.

32 B. H. Thacker, S. W. Doebling, F. M. Hemez, M. C. Anderson, J. E. Pepin, and E. A. Rodriguez. Concepts of Model Verification and Validation. Technical report, Los Alamos National Laboratory, University of California, 2004. doi:10.2172/835920.

33 U.S. G.A.O. Guidelines for Model Evaluation. Technical Report January, National Criminal Justice Reference Service (NCJRS), 1979. URL: https://www.ncjrs.gov/App/abstractdb/AbstractDBDetails.aspx?id=84469.

34 A. S. White and R. Sinclair. Quantitative validation techniques a database. (I). Simple examples. *Simulation Modelling Practice and Theory*, 12(6):451–473, 2004. doi:10.1016/j.simpat.2004.06.001.

35 T. R. Willemain. Insights on Modeling from a Dozen Experts. *Operations Research*, 42(2):213–222, 1994. doi:10.1287/opre.42.2.213.

36 M. V. Zelkowitz and D. R. Wallace. Experimental model for validating technology. *IEEE Computer*, 31(5):23–31, 1998.

Feature Extractors for Describing Vehicle Routing Problem Instances

Jussi Rasku[1], Tommi Kärkkäinen[1], and Nysret Musliu[3]

1 **Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35, FI-40014, Finland**
`jussi.rasku@jyu.fi`
2 **Department of Mathematical Information Technology, University of Jyväskylä, P.O. Box 35, FI-40014, Finland**
3 **Institute of Information Systems, Vienna University of Technology, A-1040 Vienna, Austria**

Abstract

The vehicle routing problem comes in varied forms. In addition to usual variants with diverse constraints and specialized objectives, the problem instances themselves – even from a single shared source – can be distinctly different. Heuristic, metaheuristic, and hybrid algorithms that are typically used to solve these problems are sensitive to this variation and can exhibit erratic performance when applied on new, previously unseen instances. To mitigate this, and to improve their applicability, algorithm developers often choose to expose parameters that allow customization of the algorithm behavior. Unfortunately, finding a good set of values for these parameters can be a tedious task that requires extensive experimentation and experience. By deriving descriptors for the problem classes and instances, one would be able to apply learning and adaptive methods that, when taught, can effectively exploit the idiosyncrasies of a problem instance. Furthermore, these methods can generalize from previously learnt knowledge by inferring suitable values for these parameters. As a necessary intermediate step towards this goal, we propose a set of feature extractors for vehicle routing problems. The descriptors include dimensionality of the problem; statistical descriptors of distances, demands, etc.; clusterability of the vertex locations; and measures derived using fitness landscape analysis. We show the relevancy of these features by performing clustering on classical problem instances and instance-specific algorithm configuration of vehicle routing metaheuristics.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.1.6 Optimization, G.1.10 Applications, I.2.6 Learning, I.2.8 Problem Solving, Control Methods, and Search

Keywords and phrases Metaheuristics, Vehicle Routing Problem, Feature extraction, Unsupervised learning, Automatic Algorithm Configuration

Digital Object Identifier 10.4230/OASIcs.SCOR.2016.7

1 Introduction

The quality and the required computational effort of algorithmically optimized vehicle routing solutions are heavily dependent on the problem instance, the solution method, and using the right parameters for the algorithms [5]. Fortunately, it has been shown that automatic algorithm configuration and algorithm selection can be used to improve the solver performance. Thus, in order to make routing algorithms more robust and adaptive, we propose applying machine learning to help the algorithms more effectively adapt to the problem being solved. However, as a prerequisite, we need a way to describe the problem instances to the learning algorithms.

The vehicle routing problem (VRP) can be considered to be a generalization of the traveling salesman problem (TSP). Therefore, studies where TSP instances are described, e.g. [20, 8, 12, 6, 13], are highly relevant. Smith-Miles and van Hemert [20] proposed 12 features for predicting the most suitable optimization algorithm for a TSP instance. The feature set is well-rounded containing features derived from the distance matrix, clustering, nearest neighbors, and geometry of an instance. Kanda et al. [8] had a similar goal, but they relied only on features based on the problem size and statistical description of the distance matrix, whereas Mersmann et al. [12] proposed a set of 47 features in order to build a model that could be used to discriminate between hard and easy TSP instances. Hutter et al. [6] proposed a set of new approaches such as describing minimum spanning trees, ruggedness, and probing with TSP solvers. Probing involved analyzing and describing the solution attempts with a heuristic and branch-and-cut solvers. Pihera and Musliu [13] further extended this feature set, which allowed algorithm selection for a TSP instance.

Literature of VRP descriptors is scarce. The only studies on VRP feature extraction from the machine learning perspective we are aware of are the dissertation of Steinhaus [22] and algorithm performance prediction in [25]. Steinhaus [22] explores the use of a self organizing maps in solving VRPs and in algorithm selection. She proposes 23 features specifically for VRP problems and explores the discrimination power of this feature set across 102 VRP benchmark problems. Most features she proposed are based on earlier literature on describing TSPs, but they are complemented with features describing the demand distribution of the nodes, vehicle capacity, and their relations. Studies from a VRP *fitness landscape analysis* perspective, e.g. [21, 14, 25], do exist, but as these metrics are mainly used to gain deeper understanding of the problem, they need to be adapted before they can be used for performance prediction or algorithm selection. This was the approach chosen by Ventresca et al. [25].

In this article, a set of feature extractors gathered from the aforementioned sources is adapted to describe capacitated vehicle routing problem (CVRP) instances. Our goal is to recognize problem types and better understand instance properties that may affect solving them. A set of features that is this comprehensive has not been previously used to describe vehicle routing problems. Moreover, the feature set is validated experimentally with clustering of benchmark instances, automatic algorithm configuration [5], and instance specific algorithm tuning [7].

Our contributions are threefold: First, we give a review on feature extraction of vehicle routing problems. Second, proposed features are used in automatic configuration of three metaheuristic CVRP solvers to prove their usefulness for self-adaptive and learning solution techniques. Finally, we do clustering on 168 well known CVRP benchmark instances and make observations on their similarities. To the best of our knowledge, this is the first study that proposes the use of features acquired by probing CVRP instances with exact and heuristic solution methods. This is also the first study to explore the possibility of using features to improve the performance of automatic configuration of vehicle routing algorithms.

This paper is organized as follows: In Section 2 the automatic algorithm configuration problem is defined. Section 3 introduces the vehicle routing problem in detail, with handling of common solution approaches. It is followed by listings of feature extractors and descriptors for these problems, also including those presented in this study. Section 4 describes the experimental setup and the results for verifying the proposed feature set. Finally, we conclude our study in Section 5.

2 Instance Specific Algorithm Configuration Problem

The task of automatic algorithm configuration (AAC) involves the off-line task of finding a "good" set of parameter values, or *a parameter configuration*, for a *target algorithm* in a way that the algorithm achieves the best possible performance. It is critical to use a representative set of problem instances when configuring the algorithm parameters. This ensures that the performance advantage manifests also on new, previously unseen, instances.

If a good generalized performance is needed and the problem set is not homogeneous, i.e. the instances are very different from each other, the use of AAC may even be disadvantageous: a parameter configuration may enable an algorithm to perform well on some instances, but be inferior to algorithm defaults on another. One possible solution in a situation like this is to use instance specific algorithm configuration as described e.g. in Kadioglu et al. [7]. The idea is to configure the parameters for each group of mutually similar problem instances separately, and when a new problem instance needs to be solved, the automatically configured parameters of the most similar instance group is used. For a study on instance specific algorithm configuration of a TSP metaheuristic we refer to [18].

The task of algorithm selection is closely related to AAC. In algorithm selection, the problem instance properties are used to choose the algorithm with best predicted performance. Usually the algorithm is selected out of a portfolio, and the model for algorithm performance is built earlier during an off-line learning phase. The approach has proven successful: during the last decade many state-of-the-art results in combinatorial optimization competitions have been achieved using algorithm selection from an algorithm portfolio [27, 10].

Please note that both algorithm selection and instance specific algorithm configuration need a way to describe the problem instances. Therefore, a good set of feature extractors is a critical prerequisite for employing these learning meta-optimization techniques. It is necessary to experimentally discover which features can characterize a problem set in such a way they capture properties relevant to a) solving the problems b) configuring algorithm parameters c) recognizing a set of mutually similar problems that can share a configured parameter configuration, and d) ability to predict algorithm performance.

3 The Vehicle Routing Problem

The Vehicle Routing Problem (VRP) involves finding optimal routes for *vehicles* leaving from a *depot* to serve number of *clients*. Each client must be visited exactly once by exactly one vehicle. Each vehicle must leave from the depot and return there after serving the clients on its *tour*. There are numerous variants of VRPs, each with their own additional constraints [23]. In this study, only the classic Capacitated Vehicle Routing Problem (CVRP) is considered. In the CVRP, each of the identical vehicles has a maximum carrying capacity of Q that cannot be exceeded at any point of the tour. Each of the clients, indexed with i, have a demand q_i that has to be within $0 < q_i \leq Q$. The number of vehicles, denoted by k, is the primal minimization target, followed by the total travel distance of the k vehicles. Extending this notation, the CVRP can be written in a graph formulation adapted from Toth and Vigo [23] as follows: Let $V = \{0, \ldots, n\}$ be the set of vertices where the depot has the index 0 and where the rest correspond to the clients. The size of the problem is denoted by $N = |V|$. Let $E = \{(0,1), \ldots, (i,j), \ldots, (n-1,n)\}$ be the set of edges, where $i, j \in V, i \neq j$. Therefore, the graph $G = (V, E)$ is complete with each edge $e = (i,j) \in E$ having an associated non-negative weight c_{ij} that is the cost of traversal from vertex i to j. The weights can be also given as a distance matrix D. For each of the edges $(i,j) \in E$ there is a binary decision variable x_{ij} to decide whether the edge is traversed.

3.1 On Solving Vehicle Routing Problems

The solution approaches of VRPs can be divided into two main families: exact and heuristic methods. Heuristic methods are often augmented with metaheuristics to avoid entrapment in the first local optima the search encounters. Laporte [9] further divides heuristic methods into constructive and improvement heuristics. Constructive heuristics insert unassigned clients on the routes, and improvement heuristics improve the solution quality through small moves until no improving steps can be taken. Improvement heuristics can be seen as a building block of the Local Search (LS), a key element in modern metaheuristics.

The (meta)heuristic approach is the most feasible approach when solving larger CVRP problems. However, exact methods are still relevant as the (meta)heuristic methods make no guarantees in reaching the globally optimal value. According to Lysgaard et al. [11], the most promising exact solution technique for CVRP appears to be branch-and-cut (BnC). In BnC cutting planes are iteratively added to a relaxed linear programming model to ultimately narrow down on the global optimum.

3.2 Descriptors for the Problem

The features for routing problem instances are usually calculated either using the distance matrix or the 2D coordinates. Therefore, to calculate all features, both the node coordinates and the distance matrix need to be known. If D was not given in an instance file, a distance matrix was produced using the depot and client coordinates. Likewise, if a benchmark instance provided only a distance matrix, we used multidimensional scaling (MDS) [2] to generate x and y coordinates for the instance. We also followed the example of Smith-Miles and van Hemert [20] and scaled the coordinates into the $[(0,0),(400,400)]$ rectangle to make the geometrical features comparable between problem instances. However, we retain the shape (scale) of the problem when normalizing the problem to avoid distortion of the distance matrix, i.e. we maintained the x/y ratio. To maintain the connection between the coordinates and D, we scaled the distance matrix D using the same multiplier as with coordinates. This preprocessing produces a commensurable distance matrix D^n and coordinate set P^n that can be used to calculate geometrical and graph features.

Table 1 (p. 5) presents the CVRP feature extractors used in this study. The features proposed in this study are marked with bold typeface. The table also shows how many feature values each extractor produces. Usually the features are statistical descriptors explaining the distribution of measured values. If the number of statistical descriptors is five, it includes statistical moments (mean, standard deviation, skewness and kurtosis) and coefficient of variation; whereas if 11 descriptors are given, the former are complemented by minimum, maximum, median, number of modes, frequency of the mode value, and the mode itself (or average of modes). An even more complete set of 14 descriptors adds quartiles.

Table 1a. The first feature set is for describing the node distribution on a 2D plane. The most often used feature involves statistically describing the distance matrix (cost matrix, without the diagonal). Smith-Miles and van Hemert [20] used the standard deviation, which Kanda et al. [8] and Hutter et al. [6] complemented with a more comprehensive set of statistical descriptors. We normalized the distance matrix to the rectangle $[(0,0),(400,400)]$, similarly to [20], and calculate 11 statistical descriptors for the distance distribution. Smith-Miles and van Hemert [20] also proposed counting the distinct distances found in the distance matrix using different precision. We used four levels of precision, like in [6]. Also, the centroid of the coordinates and the euclidean distance from each point to the centroid were calculated. The average of these distances is the "radius" feature from [20].

Table 1 The feature extractors for CVRPs, grouped by type.

(a) Node distribution features

ID	Feature	#
ND1	Distribution of distance matrix calues [20, 8, 6]	11
ND2	Fraction of distinct distances (with 1,2,3,4 decimals) [20, 6]	4
ND3	Centroid of the nodes (x,y) [20]	2
ND4	Distance to the centroid [20]	5
ND5	# of clusters (abs.,rel.) [20]	2
ND6	# of core, edge and outlier cluster points (rel.) [20]	3
ND7	Reach of the clusters [20]	5
ND8	Normalized cluster sizes [6]	5
ND9	**Silhouette coefficient**	1
ND10	Minimum bottleneck cost [6]	5

(b) Minimum spanning tree (MST) features

ID	Feature	#
MST1	MST edge cost [12, 6]	5
MST2	MST node degree [12, 6]	5
MST3	MST depth **from the depot**	5

(c) Local search (LS) probing features

ID	Feature	#
LSP1	Solution quality after construction phase [6]	5
LSP2	Solution quality after LS [6]	5
LSP3	Improvement per LS step [6]	5
LSP4	LS steps to local minimum [6]	5
LSP5	Distance of local minima [6]	5
LSP6	% for edges in local optima [6]	5
LSP7	Solution edge lengths per quartile (**5** $\times$ **4** quartiles) [13]	**20**
LSP8	Segment length [13]	5
LSP9	Segment edge count [13]	5
LSP10	Segment edge length [13]	5
LSP11	Intra-tour intersections [13]	5
LSP12	**Autocorrelation length**	5

(d) Branch-and-cut probing features

ID	Feature	#
BCP1	Improvement per added cut [6]	5
BCP2	Ratio between upper and lower bound [6]	1
BCP3	Solution value after probing [6]	1
BCP4	Lower bound [6]	1

(e) Geometric features

ID	Feature	#
G1	Area of the enclosing rectangle ("squareness") [20, 6]	1
G2	Convex hull (CH) area [12]	1
G3	Ratio of points on the hull [12]	1
G4	Distance of enclosed points to the CH contour [13]	11
G5	Edge lengths of the CH [13]	11

(f) Nearest neighborhood (NN) features

ID	Feature	#
NN1	Distance to 1st NN [20, 6]	5
NN2,9,15	Node input degree in directed kNN graph (DkNNG) for $k \in \{3, 5, 7\}$ [13]	14
NN3,10,16	# of strongly connected components (SCCs) in DkNNG	11
NN5,11,17	Size of SCCs in DkNNG [13]	11
NN6,12,18	# of Weakly Connected Components (WCCs) in DkNNG	11
NN7,13,19	Size of WCCs in DkNNG [13]	11
NN8,14,20	Ratio of SCCs/WCCs [13]	1
NN21	Angle between edges to two NNs [12, 13]	11
NN22	Cosine similarity between edges to two NNs [13]	11

(g) VRP specific features

ID	Feature	#
DC1	Number of clients [8]	1
DC2	The depot location (x, y) [22]	2
DC3	**Distance between the centroid and the depot**	1
DC4	**Client dist. to the depot**	5
DC5	Client Demands [22]	5
DC6	Ratio of total demand to total capacity (the "tightness") [22]	1
DC7	Ratio of max. cluster demand to vehicle capacity [22]	1
DC8	Ratio of cluster outlier to overall demand [22]	1
DC9	Ratio between the largest demand and the capacity [22]	1
DC10	Average number of clients per vehicle [22]	1
DC11	Minimum number of trucks [22]	1

Some heuristics rely heavily on the existence of clusters. Therefore, features capturing this aspect are expected to be useful in algorithm selection. DBSCAN clustering has been used, at least, in [20, 8, 12, 6, 13, 22] to extract features for routing problem instances. We calculated features for cluster count (absolute and relative to N); cluster size; and a relative number of core, edge and outlier points. Mersmann et al. [12] used three different values for the ϵ (maximum allowed distance for two points belonging to a same cluster), while Steinhaus [22] experimented with four alternative methods to find a good ϵ value. In our study we decided to use the minimum cluster size of 4, and with that $\epsilon = A_{\text{bb}}/(\sqrt{N}-1)$, which is an approximation of the 4th nearest neighbor distance if the nodes are assumed to be uniformly distributed on a lattice within a square with an area of A_{bb} [22]. To include a feature measuring the quality of the DBSCAN clustering, we propose the silhouette score [19] as a novel addition to the feature set.

The node distribution features are completed with the Minimum Bottleneck Cost (MBC) as proposed by [6]. It is used to describe the clusterability of TSP instances. The bottleneck cost is defined to be the weight of the longest edge on a path from node i to j, $i \neq j$. We get the minimum bottleneck cost by taking a minimum of bottleneck costs over all possible paths from node i to j. By calculating the bottleneck cost for all possible node pairs $i, j \in V$, $i \neq j$, we get a distribution that can then be described with statistical moments.

Table 1b. A minimum spanning tree (MST) was calculated for the fully connected normalized graph G^n. As suggested in [12, 6], the distribution of edge costs and node degrees of the MST were described using statistical moments. Mersmann et al. [12] included the spanning tree node depth as well, which we adapted for the VRP by calculating it with the depot as the root. We omitted the sum of the MST tree cost proposed in [12], as it can be inferred from the average MST cost.

Table 1c Probing features are computed with a solution attempt on a problem instance. An algorithm is run for predefined time or steps and the trajectory of the search is recorded. The approach is general and applicable to a variety of problems. Probing has been shown to be useful, e.g. for predicting the performance of an algorithm [6].

To adapt the TSP LS probing features from [13], we used the VRPH heuristic search algorithm library [3], or more specifically, its `vrp_init` application that is based on the Clarke-Wright construction heuristic. It was modified to accept a shape parameter γ that affects the savings calculation [28]. The parameter can be selected randomly to produce varied initial solutions. After construction, the solution is improved with intra-route multi-neighborhood search using best accept strategy with one-point-move, two-point-move and two-opt local search heuristics [3] until no improving move is found.

By repeating the probing 20 times, we could calculate the statistical descriptors in Table 1c. Some of the features closely resemble those we have used previously to validate visualization technique for VRPs with solution space analysis (SSA) [16]: the first is the distribution of Manhattan distances between the local optimum solutions (`LSP5`), calculated from the differences in edge traversal decision variable values between the solutions. This feature is a measure of the multimodality and indicator for the existence of a "big-valley" structure [14]. The second SSA feature `LSP6` describes the distribution of probabilities of all edges in locally optimal solutions, which aims to reveal the existence of a backbone [26], that is, a common structure between good solutions.

The features `LSP8-10` involve the concept of a segment. A segment is a continuous path of consecutive edges on a tour, from which the longest edges are removed as specified in [13]. Pihera and Musliu [13] also proposed another extension to the set of local search

features, which is the number of intra-tour intersections, i.e. the times the edges of a tour cross each other.

The final local search probing feature is the autocorrelation length λ_{ACL} of a random walk through a series of best-accept one-point-move neighborhoods (heuristic described e.g. in [3]). This closely relates to the autocorrelation coefficient used in [6]. For calculating the autocorrelation length we used a method adapted from [21] and [4], with a random walk length of $2N$. The walk is repeated and the length calculated 10 times.

Table 1d. Besides heuristics, also exact solvers can be used to probe the problem. We used the open source mixed-integer programming package SYMPHONY 5.6.15 and its VRP application that can solve CVRPs [15]. Unfortunately, we were not able to compile the VRP application with heuristics support for this version of SYMPHONY. Therefore, the upper bound is set only after the first feasible solution is found. Because of this, is possible that the feature `BCP2` is left undefined for the larger instances if no feasible solution is found. SYMPHONY also requires the number of vehicles k as an input when solving an instance. If k was not known we divided the total demand with some margin (+5%) with the vehicle capacity Q to get the value for k, i.e. $k = \lceil 1.05 \sum q_i/Q \rceil$. For branch-and-cut probing we used a wall time cutoff of 3.0 seconds.

Table 1e. Geometric features try to capture information of the overall shape of the problem. The area of an enclosing rectangle, when normalized with the area of the scaled problem, describes the "squareness" of the problem. Mersmann et al. [12] suggested two features concerning the convex hull: the hull area and the fraction of nodes that are on the hull contour. According to their experiments, convex hull features allow accurate separation of easy and hard TSP instances. Pihera and Musliu [13] added statistical descriptors for distances of inner nodes to the hull contour. It is assumed that the more evenly distributed the nodes are inside the convex hull, the more difficult it is to solve. Therefore, all these were included in our feature set.

Table 1f. Because heuristic solution methods operate by navigating through the search space using a local search neighborhood, the Nearest Neighbors (NN) of the nodes can offer important insight to the structure of the problem. In our study, the distribution of the 1st nearest neighbor distances over all nodes is statistically described as done in [20, 6, 13]. We also included the extended nearest neighbor features presented in [13], which involve building a directed graph by taking only $k \in 3, 5, 7$ shortest edges for each node from the complete normalized graph G^n. The node degree, number and size of strongly and weakly connected components, and their ratios are calculated and statistically described.

Table 1g. In describing the demands and capacity, we followed [22]. As an extension to the VRP specific features, we propose measuring the distance between the depot and the centroid of the client points. Also, describing the shape of the distribution of distances from clients to the depot is included in our feature set. Furthermore, the size of the problem (number of clients) is included here. Refer to [22] for details on these features.

In addition to the features presented in Table 1, we recorded the per instance feature computation time as proposed in [6]. These are reported as timing features `T1-T9` that match the feature groups (Tables 1a–1g), with the exception of the autocorrelation and bottleneck cost features, which are timed separately.

To summarize this section, we have adapted and proposed 76 feature extractors for CVRPs which generate 386 features in total. The feature extractors were implemented in Python version 2.7.10, with the aid from numerical libary Numpy (version 1.9.2), machine learning library Scikit-learn (version 0.16.1), and statistical library Scipy (0.15.1). VRPH and SYMPHONY were built with GNU g++ 5.3.0 compiler from the Mingw-w64 project.

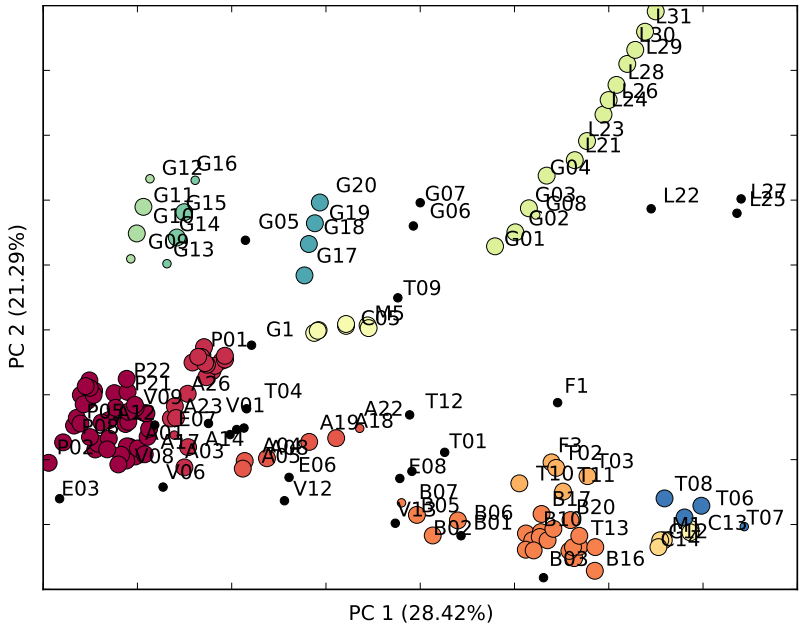

Figure 1 The clustering of the benchmark instances. Black dots are non-clustered outliers. The plot axes are the first two principal components, with the ratio of explained variance given in parenthesis.

All feature extraction in this study was done on a laptop with dual-core 2.53 GHz Intel Core i5 520M processor, 8 GB of memory and 64-bit Windows 7 Enterprise operating system.

4 Experimental Evaluation of the Features

4.1 Clustering

To evaluate the quality of the proposed feature set, we computed the 386 features for each of the 168 problem instances in CVRPLIB, which is a collection of CVRP benchmark instances [24]. However, the high dimensionality of the resulting data had to be addressed before clustering. Hutter et al. [5] suggests using principal component analysis (PCA) to reduce the computational complexity when building a surrogate model for the automatic algorithm configuration tool SMAC. We share some of the concerns regarding the computational cost. However, in our case a larger issue is the curse of dimensionality, where the space volume grows very rapidly as the dimensionality increases. This makes the dataset too sparse to provide a representative sample of the high dimensional space. A related problem is the irrelevancy of the distance metric in high dimensional data, where all data points seem to be similarly close to each other [1]. To overcome these issues, we reduced the dimensionality of the feature space with PCA.

To do the actual clustering, the feature data was first normalized by scaling all features independently to a range $[0.0, 1.0]$. Then, PCA was applied to bring the dimensionality of the data down from 386 to 7 following the example of [5]. These seven principal components together explain 71 % of the overall variance in the data. Finally, to do the unsupervised learning, we used the DBSCAN with a minimum cluster size of 3. The ϵ parameter was set to 0.20 through experimentation. Resulting clusters for the 168 benchmark instances are presented in Figure 1.

The clusters in the lower left corner seem to be the small-to-medium easy-to-solve instances. Unsurprisingly, the A and P sets overlap, as P is based on A. The difference between A and B is that clients in A are uniformly generated whereas in B they are clustered. Also the Taillard (T) and Fisher (F) sets contain clustered clients, which can be observed as an overlap with the set B. Interestingly, the benchmark set Golden (G) is separated into four clusters and some outliers. The benchmark set contains points in geometric shapes like stars, squares, circles and rays, and it seems that our features are able to discriminate between these. The Li (L) and Golden benchmark sets are similar and clustering them together is expected. For a more accurate analysis of clusters we would need to do a more extensive experimentation with the solvers, since probing does not necessarily allow reliable estimation of the hardness and computational difficulty of an instance. The complete list of problem instance abbreviations and the clustering in a table format, together with an interactive zooming visualization of the clustering, can be found from the supplementary online appendix at `http://users.jyu.fi/~juherask/features/`

4.2 Instance Specific Algorithm Configuration

As the automatic algorithm configuration targets, we used the three metaheuristic solvers provided by the VRPH package from Groër et al. [3]. Each solver employs different metaheuristic: Record-to-Record travel (VRPH-RTR, 6+8 free parameters), simulated annealing (VRPH-SA, 6+5), and ejection (VRPH-EJ, 6+3). We omit the descriptions of the algorithms and solver parameters and refer the reader to [3], whereas a detailed description of the automatic algorithm configuration setup can be found in [17].

As a configurator, we used SMAC [5]. SMAC is a state-of-the-art AAC method that alternates between fitting a random forest model to the observed behavior of the target algorithm, and using that model to predict the performance of generated parameter configuration candidates – evaluating only those that are most promising on the solver. SMAC offers an option to complement the problem instances with feature values, which are used when building and updating the random forest model. In our experiments this approach is called fSMAC. fSMAC already does PCA to the feature vectors, but as an additional preprocessing step we took 50 features that showed the highest correlation with the solution quality in heuristic and branch-and-bound probing. SMAC is not an instance specific algorithm configuration tool like ISAC from Kadioglu et al. [7], but we can follow a similar scheme to create IS-fSMAC. This variant uses k-Means clustering on the preprocessed feature data to split the problem instance set to subsets. These subsets supposedly share similar solving characteristics and can be configured separately.

In our configuration experiments we used a set of 14 instances taken with stratified sampling from the CVRPLIB set A. The problem set is the same one that we used in [17], which makes it possible for the interested reader to compare the proposed approach against other configurators. Also, each configuration task was run with three different evaluation budgets (EBs): 100, 500, and 1000 time capped (10 s) runs of the target algorithm. In the case of configuring the clustered instances, the budget was distributed according to the cluster size. Because the algorithms are stochastic, the experiments were repeated 10 times.

The results of the configuration tasks are presented in Table 2. The use of features seems beneficial, especially with a budget of 100. This is unsurprising, as the use of features is expected to provide more initial information when building the surrogate model of the parameter-solution quality response surface. Especially VRPH-SA target seems to benefit from using the features. The advantage gained by using features is smaller for VRPH-EJ and VRPH-RTR targets, but the effect still exists. However, the results of instance specific

Table 2 Median tuning results for VRPH metaheuristics. Results are given as percentages from the aggregated best known solution (relative optimality gap). The best known solution values are from CVRPLIB. Statistically better results are in bold ($p < 0.05$ with Bonferroni adjustment). If no single best was found, a test for a best pair was made.

Target	**VRPH-SA**			**VRPH-EJ**			**VRPH-RTR**		
Defaults	0.83 (0.12)			0.50 (0.13)			1.42 (0.07)		
Method \ EB	**100**	**500**	**1000**	**100**	**500**	**1000**	**100**	**500**	**1000**
SMAC	**0.40**	0.29	0.26	0.39	**0.35**	0.34	**0.16**	0.09	0.10
	(0.11)	(0.08)	(0.06)	(0.07)	**(0.04)**	(0.04)	**(0.05)**	(0.01)	(0.02)
fSMAC	**0.39**	0.26	**0.23**	**0.36**	0.36	0.35	**0.15**	0.09	**0.07**
	(0.15)	(0.06)	**(0.05)**	**(0.06)**	(0.05)	(0.06)	**(0.06)**	(0.04)	**(0.03)**
IS-fSMAC	0.56	0.27	0.25	**0.35**	**0.33**	0.34	0.19	0.09	0.09
	(0.11)	(0.07)	(0.05)	**(0.08)**	**(0.07)**	(0.06)	(0.06)	(0.03)	(0.01)

parameter tuning are not as good as expected. The clustering to problem classes seems to be beneficial only for VRPH-EJ targets. It may be that the features are unable to capture the differences (unlikely), the clustering is handicapped by the curse of dimensionality (likely), or the evaluation budget split among clusters is too small for SMAC to converge to good parameter configurations (likely). Still, the most probable cause is the homogeneity of the problem set. All of the instances in the set A come from the same generator, thus showing similar solving characteristics. Additional experiments are needed to identify the largest factor preventing the instance specific tuning from giving comparable advantage to what has been reported in e.g. in [7]. Nonetheless, every resulting parameter configuration is superior compared to the defaults.

All automatic configuration was done on a computing server with 64 Intel(R) Xeon(R) CPU E7 2.67 GHz cores, and 1 TB of RAM running 64-bit OpenSUSE version 13.2 (codename Harlequin). We enforced a 10 second cutoff for all evaluations of the CVRP solver.

5 Conclusions

In this article, we set out to find feature extractors for capacitated vehicle routing problem (CVRP) instances, mostly by adapting Traveling Salesman Problem (TSP) descriptors from the literature. We implemented 76 feature extractors for almost every descriptor that had been reportedly used in algorithm selection and automatic algorithm configuration of routing algorithms and proposed some novel ones. The presented set of 386 features for CVRP is unparalleled in its extent. Additionally, we are not aware that probing with heuristic and branch-and-cut solvers has been previously used to produce features for CVRP meta-optimization. The suitability of these features was verified with feature assisted automatic algorithm configuration with the state-of-the-art tool SMAC. We also presented clustering of 168 well-known benchmark instances from the CVRPLIB collection. Clustering shows good discrimination ability between the known properties of these problems. However, a more complete analysis of the clustering is warranted to get novel insights.

We can conclude that automatic algorithm configuration can benefit from using the proposed features. Out of the tested CVRP metaheuristics, the simulated annealing (VRPH-SA) benefited the most. We also experimented with instance specific configuration, where it was possible to further improve the configured solver performance of the ejection metaheuristic (VRPH-EJ). However, the overall increase in performance when using an instance specific

algorithm configuration scheme was modest. This is probably due to our problem set being relatively small and homogeneous. Therefore, a more extensive experimentation with different targets, instance specific configurators, and problem sets is required to make a judgment on applicability of instance specific parameter configuration of vehicle routing solvers. Also, please note that in our automatic algorithm configuration experiments we did not test for over-tuning (cf. overfitting), which may manifest as poor generalizability of the configured parameter configuration.

Other future research topics include: Feature selection that should help us recognize the most useful features, as currently the high dimensionality of the feature vector seems to confuse unsupervised learning and algorithm configuration efforts. We would also like to extend our feature extractors to describe other well-known VRP variants such as vehicle routing problem with time windows (VRPTW) and pickup and delivery problems (PDP). This could potentially reveal new interesting similarities between the problem types and sets. We would also like to extend our study towards algorithm selection.

It has been shown that applying feature based machine learning approaches, such as the one presented here, in solving combinatorial optimization problems, can lead to significant improvements in on-line algorithm performance and resulting solution quality. Adapting this approach in solving VRPs has shown promise and warrants further research.

References

1 Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional space. In Jan Bussche and Victor Vianu, editors, *Proceedings of Database Theory (ICDT 2001): 8th International Conference*, pages 420–434, Berlin, Heidelberg, 2001. Springer.

2 I Borg and P Groenen. Modern multidimensional scaling: theory and applications. *Journal of Educational Measurement*, 40(3):277–280, 2003.

3 Chris Groër, Bruce Golden, and Edward Wasil. A library of local search heuristics for the vehicle routing problem. *Mathematical Programming Computation*, 2(2):79–101, April 2010.

4 Wim Hordijk. A measure of landscapes. *Evolutionary computation*, 4(4):335–360, 1996.

5 Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization*, pages 507–523. Springer, 2011.

6 Frank Hutter, Lin Xu, Holger H Hoos, and Kevin Leyton-Brown. Algorithm runtime prediction: Methods & evaluation. *Artificial Intelligence*, 206:79–111, 2014.

7 Serdar Kadioglu, Yuri Malitsky, Meinolf Sellmann, and Kevin Tierney. ISAC – instance-specific algorithm configuration. In Helder Coelho, Rudi Studer, and Michael Wooldridge, editors, *Proceedings of 19th European Conference on Artificial Intelligence (ECAI 2010*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 751–756. IOS Press, 2010.

8 Jorge Kanda, Andre Carvalho, Eduardo Hruschka, and Carlos Soares. Selection of algorithms to solve traveling salesman problems using meta-learning. *International Journal of Hybrid Intelligent Systems*, 8(3):117–128, 2011.

9 Gilbert Laporte. What you should know about the vehicle routing problem. *Naval Research Logistics*, 54(8):811–819, 2007.

10 Marius Lindauer, Holger H. Hoos, Frank Hutter, and Torsten Schaub. AutoFolio: An automatically configured algorithm selector. *Journal of Artificial Intelligence Research*, 53:745–778, 2015.

11 Jens Lysgaard, Adam N Letchford, and Richard W Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2):423–445, 2004.

12 Olaf Mersmann, Bernd Bischl, Heike Trautmann, Markus Wagner, Jakob Bossek, and Frank Neumann. A novel feature-based approach to characterize algorithm performance for the traveling salesperson problem. *Annals of Mathematics and Artificial Intelligence*, 69(2):151–182, 2013.

13 Josef Pihera and Nysret Musliu. Application of machine learning to algorithm selection for TSP. In *Tools with Artificial Intelligence (ICTAI), IEEE 26th International Conference on*, pages 47–54. IEEE, 2014.

14 E. Pitzer, S. Vonolfen, A. Beham, M. Affenzeller, V. Bolshakov, and G. Merkuryeva. Structural analysis of vehicle routing problems using general fitness landscape analysis and problem specific measures. In *14th International Asia Pacific Conference on Computer Aided System Theory*, pages 36—38, 2012.

15 Ted K Ralphs. Parallel branch and cut for capacitated vehicle routing. *Parallel Computing*, 29(5):607–629, 2003.

16 Jussi Rasku, Tommi Kärkkäinen, and Pekka Hotokka. Solution space visualization as a tool for vehicle routing algorithm development. In Mikael Collan, Jari Hämälainen, and Pasi Luukka, editors, *Proceedings of the Finnish Operations Research Society 40th Anniversary Workshop (FORS40)*, volume 13, pages 9–12. LUT Scientific and Expertise Publications, 2013.

17 Jussi Rasku, Nysret Musliu, and Tommi Kärkkäinen. Automating the parameter selection in VRP: an off-line parameter tuning tool comparison. In William Fitzgibbon, A. Yuri Kuznetsov, Pekka Neittaanmäki, and Olivier Pironneau, editors, *Modeling, Simulation and Optimization for Science and Technology*, pages 191–209. Springer, 2014.

18 Jana Ries, Patrick Beullens, and David Salt. Instance-specific multi-objective parameter tuning based on fuzzy logic. *European Journal of Operational Research*, 218(2):305–315, 2012.

19 Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

20 Kate Smith-Miles and Jano van Hemert. Discovering the suitability of optimisation algorithms by learning from evolved instances. *Annals of Mathematics and Artificial Intelligence*, 61(2):87–104, 2011.

21 Peter F Stadler. Landscapes and their correlation functions. *Journal of Mathematical Chemistry*, 20(1):1–45, 1996.

22 Meghan Steinhaus. *The Application of the Self Organizing Map to the Vehicle Routing Problem.* PhD thesis, University of Rhode Island, 2015.

23 Paolo Toth and Daniele Vigo, editors. *The vehicle routing problem.* SIAM, 2002.

24 E. Uchoa, D. Pecin, A. Pessoa, M. Poggi, A. Subramanian, and T. Vidal. New benchmark instances for the capacitated vehicle routing problem. Technical report, UFF, Rio de Janeiro, Brazil, 2014.

25 Mario Ventresca, Beatrice Ombuki-Berman, and Andrew Runka. Predicting genetic algorithm performance on the vehicle routing problem using information theoretic landscape measures. In Martin Middendorf and Christian Blum, editors, *Proceedings of the Evolutionary Computation in Combinatorial Optimization: 13th European Conference (EvoCOP 2013)*, pages 214–225, Berlin, Heidelberg, 2013. Springer.

26 Toby Walsh and John Slaney. Backbones in optimization and approximation. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01)*, 2001.

27 Lin Xu and Kevin Leyton-brown. SATzilla : Portfolio-based algorithm selection for SAT. *Artificial Intelligence*, 32:565–606, 2008.

28 P.C. Yellow. A computational modification to the savings method of vehicle scheduling. *Operational Research Quarterly*, 21:281–293, 1970.

Towards the Integration of Power-Indexed Formulations in Multi-Architecture Connected Facility Location Problems for the Optimal Design of Hybrid Fiber-Wireless Access Networks*

Fabio D'Andreagiovanni[1], Fabian Mett[2], and Jonad Pulaj[3]

1 **Department of Mathematical Optimization, Zuse Institute Berlin (ZIB), Takustraße 7, 14195 Berlin, Germany; and DFG Research Center MATHEON and Einstein Center for Mathematics, Straße des 17 Juni, 10623 Berlin, Germany; and Institute for System Analysis and Computer Science, National Research Council of Italy (IASI-CNR), via dei Taurini 19, 00185 Roma, Italy**
d.andreagiovanni, mett, pulaj@zib.de

2 **Department of Mathematical Optimization, Zuse Institute Berlin (ZIB), Takustraße 7, 14195 Berlin, Germany**
mett@zib.de

3 **Department of Mathematical Optimization, Zuse Institute Berlin (ZIB), Takustraße 7, 14195 Berlin, Germany; and DFG Research Center MATHEON and Einstein Center for Mathematics, Straße des 17 Juni, 10623 Berlin, Germany**
pulaj@zib.de

Abstract

Urban access networks are the external part of worldwide networks that make telecommunication services accessible to end users and represent a critical part of the infrastructures of modern cities. An important recent trend in urban access networks is the integration of fiber and wireless networks, leading to so-called fiber-wireless (Fi-Wi) networks. Fi-Wi networks get the best of both technologies, namely the high capacity offered by optical fiber networks and the mobility and ubiquity offered by wireless networks. The optimal design of fiber and wireless networks has been separately extensively studied. However, there is still a lack of mathematical models and algorithms for the integrated design problem. In this work, we propose a new Power-Indexed optimization model for the 3-architecture Connected Facility Location Problem arising in the design of urban telecommunication access networks. The new model includes additional power-indexed variables and constraints to represent the signal-to-interference formulas expressing wireless signal coverage. To solve the problem, which can prove very hard even for a state-of-the art optimization solver, we propose a new heuristic that combines a probabilistic variable fixing procedure, guided by (tight) linear relaxations, with an MIP heuristic, corresponding to an exact very large neighborhood search. Computational experiments on realistic instances show that our heuristic can find solutions of much higher quality than a state-of-the-art solver.

1998 ACM Subject Classification G.1.6 Optimization

* The work of Fabio D'Andreagiovanni and Jonad Pulaj was partially supported by the *Einstein Center for Mathematics Berlin* (ECMath) through Project MI4 (ROUAN) and by the *German Federal Ministry of Education and Research* (BMBF) through Project VINO (Grant 05M13ZAC) and Project *ROBUKOM* (Grant 05M10ZAA).

5th Student Conference on Operational Research (SCOR'16).
Editors: Bradley Hardy, Abroon Qazi, and Stefan Ravizza; Article No. 8; pp. 8:1–8:11

Open Access Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Keywords and phrases Telecommunications Access Networks, Connected Facility Location, Mixed Integer Linear Programming, Power-Indexed Formulations, MIP Heuristics

Digital Object Identifier 10.4230/OASIcs.SCOR.2016.8

1 Introduction

The volume of data exchanged over telecommunications networks has enormously increased in the last two decades and telecommunication companies predict that such increase will relentlessly continue. This has originated the need for more technologically advanced and complex telecommunications networks. Within this context, *access networks*, namely the "external" part of a telecommunication network that connects users to their service providers, have experienced a deep technological evolution and have become a vital part of modern smart cities. Last generation access networks heavily rely on the use of optical fiber connections, which provide much higher capacity and better transmission rates than the traditional copper-based connections. Since the deployment of a pure optical fiber access network is nowadays considered impractical and uneconomical, in recent times, different types of hybrid optical fiber deployments have been proposed to provide broadband access. Taken as a whole, these several deployments, usually called *architectures*, are commonly referred to by the acronym *FTTX* (Fiber-To-The-X): here, the *X* specifies to which point of the network the optical fiber is brought. Major examples of architectures are: *Fiber-To-The-Home (FTTH)*, which brings a fiber directly to the final user; *Fiber-To-The-Cabinet (FFTC)* and *Fiber-To-The-Building* (FTTB), which bring a fiber to a street cabinet or to the building of the user, respectively (the fiber termination point is then connected to the user typically through a copper-based connection). We refer the reader to [11] for an exhaustive introduction to FTTX networks and their design. A recent and promising trend in FTTX has been represented by the integration of wired and wireless connections, leading to *3-architecture* networks that include also the so-called *Fiber-To-The-Air* (FTTA) architecture [10, 11]. Such 3-architecture represents an evolution of mixed-wired *2-architecture* networks like FTTH and FTTC/FTTB (see e.g., [14]). A 3-architecture network is aimed at getting the best of both wired and wireless worlds: the high capacity offered by optical fiber networks and the mobility and ubiquity offered by wireless networks [10]. Additionally, it grants a determinant cost advantage, since deploying wireless transmitters is cheaper and faster than deploying optical fibers, which requires costly and time-consuming excavations.

In this paper, we present a new optimization model based on Power-Indexed formulations for the design of 3-architecture access networks that integrate wired fiber/copper connections with wireless connections. With respect to state-of-the-art literature (we refer the reader to [11] and [14] for an overview), our model has the merit of including the formulas that are recommended by international telecommunications regulatory bodies to evaluate service coverage in wireless networks. Such formulas are the Signal-to-Interference Ratios (SIRs) [15], which evaluate the strength of the wireless signal providing service with respect to the total strength of the wireless interfering signals. The inclusion of SIRs is critical in wireless network design problem that consider wireless signal coverage: their exclusion may indeed lead to wrong design solutions (see [6, 7] for a discussion). This work represents also a refinement of the first study that we made in [4] and that we improve here by using a more-advanced power-indexed model for wireless network design [6]. In this work, our main original contributions are:

1. we propose a power-indexed formulation for optimally designing a 3-architecture access network, modelling the *signal-to-interference formulas* that express wireless signal coverage through discrete power emission decision variables;
2. we strengthen the basic power-indexed formulation of the problem by using a set of tight valid inequalities that model forbidden power configurations of the wireless transmitters;
3. since the problem can result difficult even for a state-of-the-art MIP solver, we propose to solve it by a heuristic based on the combination of a probabilistic procedure for fixing variables, guided by Linear Programming (LP) relaxations of the problem, with a Mixed Integer Programming (MIP) heuristic, which executes an *exact very large neighborhood search* (by the term "exact", we mean that the search is formulated as an MIP problem that is then solved exactly by an MIP solver);
4. we present computational results obtained for realistic network instances, showing that our new algorithm can return solutions of much higher quality than those provided by a state-of-the-art MIP solver.

2 A Power-Indexed model for 3-architecture access networks

In order to derive a power-indexed formulation for hybrid fiber-wireless network design, we first need to define a generalization of a *Connected Facility Location Problem* (ConFL) that includes three types of architectures. For a thorough introduction to concepts of graph and network flow theory and to the ConFL, we refer the reader to the book [1] and to the paper [12]. Given a set of users and a set of openable facilities that may serve the users, we can essentially describe the ConFL as the problem of deciding: (a) which facilities to open; (b) how to assign served users to open facilities; (c) how to connect open facilities through a Steiner tree; in order to minimize the total cost deriving from opening and connecting facilities and the assignment of facilities to users. The canonical ConFL considers a *single* network architecture and has been introduced and proven to be NP-Hard in [13].

A 3-architecture ConFL (3-ConFL) representing a network integrating fiber, copper and wireless technologies can be obtained by properly generalizing a 2-architecture version of the ConFL, which has been first introduced in [14]. The 3-ConFL associated with access network design involves a set of potential telecommunications facilities that can provide services to a set of potential users by installing one of the three available technologies. Each facility that is opened must be connected to a central office and each served user must be assigned to exactly one open facility. The objective of the design problem is to minimize the total cost of deployment of the network, while guaranteeing a minimum user coverage by each technology.

We denote the set of available technologies by $T = \{1, 2, 3\}$ and conventionally we assume that $t = 1$ is the optical fiber technology, $t = 2$ the copper technology and $t = 3$ the wireless technology. As first step to derive an optimization model, we introduce a directed graph $G(V, A)$ to model the network. In $G(V, A)$:

- the set of nodes V corresponds to the disjoint union of:
  1. a set of users U - each user $u \in U$ is associated with a weight $w_u \geq 0$ expressing its importance;
  2. a set of facilities F - each facility $f \in F$ can be opened at a cost $c_f^t \geq 0$ that depends upon the technology $t \in T$ that it installs;
  3. a set of central offices Γ - each office $\gamma \in \Gamma$ can be opened at a cost $c_\gamma \geq 0$;
  4. a set of Steiner nodes S.

  We call *core nodes* the subset of nodes $V^C = F \cup \Gamma \cup S$ that does not include the user nodes. Additionally, we denote by F_u^t the subset of facilities using technology t that may

serve user u and by U_f^t the subset of users that may be served by facility f when using technology t. We also denote by $F_u = \cup_{t \in T}$ the set of all the facilities that can serve u;

- the set of arcs A is the disjoint union of:
  1. a set of *core arcs* $A^{\mathrm{C}} = \{(i,j) : i,j \in V^{\mathrm{C}}\}$ that represent connections only between core nodes and are associated with a cost of realization $c_{ij} \geq 0$;
  2. a set of *assignment arcs* $A^{\mathrm{ASS}} = \{(f,u) \in A : u \in U, f \in F_u\}$ representing connection of facilities to users and associated with a cost of realization c_{fu}^t that depends upon the used technology.

We call *core graph* the subgraph $G^{\mathrm{C}}(V^{\mathrm{C}}, A^{\mathrm{C}})$ of $G(V,A)$ that represents the potential topology of the *core network*, namely the fiber-based network that interconnects the facilities and the central offices. In order to consider the cost of opening central offices in the optimization model, we adopt the modeling expedient of adding an artificial root node r to $G(V,A)$. We then introduce a set of (artificial) *root arcs* $A^{\mathrm{R}} = \{(r,\gamma) : \gamma \in \Gamma\}$ to represent the connection of the root node to every central office $\gamma \in \Gamma$. Each arc $(r,\gamma) \in A^{\mathrm{R}}$ has a cost $c_{r\gamma}$ set equal to the cost c_γ of opening the office γ. The set A^{R} is included in $G(V,A)$ and we use the notation $A^{\text{R-C}} = A^{\mathrm{R}} \cup A^{\mathrm{C}}$ to denote the union of the root and the core arcs.

One requirement in the design problem is to guarantee a minimum weighted coverage of users for each architecture. Specifically, with the total weight of users denoted by $W = \sum_{u \in U} w_u$, we express the coverage requirement for technology $t \in T$ by defining thresholds $W_t \in [0, W]$, $t \in T$. The total cost of a design solution of the access network is equal to the sum of the cost of opening central offices and facilities, the cost of connections activated in the core graph and the cost of connecting open facilities to served users.

Modeling wireless coverage. Until now, we have introduced all the elements that allow us to define an optimization model for 3-ConFL that does not include the formulas used to assess wireless coverage. In order to include such formulas, we must first briefly discuss basic concepts from wireless network design related to configuring wireless transmitters. For an introduction to the concepts of wireless network design, we refer the reader to [6, 15]. In our case, a wireless transmitter is a facility installing the technology $t = 3$. Each wireless transmitter is characterized by a number of radio-electrical parameters to set (e.g., the power emission, the tilt of the antenna and the frequency used to transmit). All these parameters could be in principle set in an optimal way, by solving an appropriate mathematical optimization problem, but in practice it is typical to optimize just a subset of them [5, 6, 9]. The vast majority of the models available in literature includes the setting of power emissions of the transmitters, since these are critical parameters that deeply affect the service coverage of the users. Such power emissions are commonly modelled by semi-continuous power variable. However, as shown in [6], it is better to consider a set of discrete power values both from a theoretical and an applied point of view: we can indeed derive effective (strong) valid inequalities and be in line with the practice of professionals, who commonly consider a (small) set of discrete power values for each transmitter. In order to model power emissions in a range $[P_{\min}, P_{\max}]$, we thus introduce a set of discrete power values $\mathcal{P} = \{P_1, \ldots, P_{|\mathcal{P}|}\}$, with $P_1 = P_{\min}$ and $P_{|\mathcal{P}|} = P_{\max}$ and $P_i > P_{i-1}$, for $i = 2, \ldots, |\mathcal{P}|$. Then, for each $f \in F$, we introduce one binary variable φ_{fl} (*power variable*) that is equal to 1 if f emits power P_l and 0 otherwise. The power emitted by a facility f can be thus denoted by $p_f = \sum_{l \in L} P_l \, \varphi_{fl}$, where $L = \{1, \ldots, |\mathcal{P}|\}$ is the set of power value indices or simply *power levels* (we must then add the constraint that the emission of f can be a single power value).

Every user $u \in U$ may pick up signals from each facility $f \in F$ installing a wireless transmitter and the power P_{fu} that u gets from f is proportional to the emitted power p_f

by a factor $a_{fu} \in [0,1]$, i.e. $P_{fu} = a_{fu}\, p_f$. The factor a_{fu} is a coefficient that summarizes the reduction in power experienced by a signal propagating from f to u [15]. A user $u \in U$ is said *covered* or *served* if it receives the wireless service signal within a minimum level of quality. The service is provided by one single wireless facility, chosen as *server* of the user, while all the other wireless facilities interfere with the server and reduce the quality of service. The minimum quality condition can be expressed through the *Signal-to-Interference Ratio (SIR)*, a measure comparing the power received from the server with the sum of the power received by the interfering transmitters [15]:

$$\frac{a_{fu}\ \left(\sum_{l\in L} P_l\ \varphi_{fl}\right)}{N + \sum_{k\in F\setminus\{f\}} a_{ku}\ \left(\sum_{l\in L} P_l\ \varphi_{kl}\right)}\ \geq\ \delta\ . \tag{1}$$

The user is served if the SIR is at least equal to a threshold $\delta > 0$ that expresses the minimum wanted quality of service. In the denominator, the coefficient $N > 0$ represents the noise of the system. The inequality (1) can be reorganized by simple operations in the so-called *SIR inequality*: $a_{fu}\ \left(\sum_{l\in L} P_l\ \varphi_{fl}\right) - \delta \sum_{k\in F\setminus\{f\}} a_{ku}\ \left(\sum_{l\in L} P_l\ \varphi_{kl}\right)\ \geq\ \delta\ N$.

Deciding which wireless facility $f \in F$ is the server of some user $u \in U$ is part of the decision process. As a consequence, we must activate or deactivate the SIR inequalities depending upon the wireless facility-user assignment. We thus face a disjunction of constraints, which we can model by modifying the SIR inequality. To this end, we must first define the set of *assignment arc variables* $y^t_{fu} \in \{0,1\}\ \forall (f,u) \in A^{\text{ASS}}\ \forall u \in U,\ f \in F^t_u,\ t \in T$: the generic variable y^t_{fu} is equal to 1 if facility f is connected to user u by technology t and is 0 otherwise. Using the assignment variable y^3_{fu}, representing the service connection of u through facility f by the wireless technology $t = 3$, and by defining a sufficiently large positive constant M (the so-called *big-M coefficient*), we define the modified SIR constraint:

$$a_{fu}\left(\sum_{l\in L} P_l\ \varphi_{fl}\right) - \delta \sum_{k\in F\setminus\{f\}} a_{ku}\left(\sum_{l\in L} P_l\ \varphi_{kl}\right) + M(1 - y^3_{fu}) \geq \delta N \tag{2}$$

It is straightforward to check that if $y^3_{fu} = 1$, then u is served by f through wireless technology and (2) reduces to a SIR inequality to satisfy. On the contrary, if $y^3_{fu} = 0$, then M activates, thus making (2) satisfied by any power configuration and therefore redundant.

Putting together all the elements that we have introduced, we can finally define a *Mixed Integer Linear Programming* (MILP) problem for modelling the 3-ConFL. To this end, we first introduce the following additional families of variables:

1. facility opening variables $z^t_f \in \{0,1\}\ \forall f \in F, t \in T$ (z^t_f is equal to 1 if facility f is open and uses technology t and is 0 otherwise);
2. arc installation variables $x_{ij} \in \{0,1\}\ \forall (i,j) \in A^{\text{R-C}}$ (x_{ij} is equal to 1 if the root or core arc (i,j) is installed and is 0 otherwise);
3. user variables $v^t_u \in \{0,1\}, \forall u \in U,\ t \in T$ (v^t_u is equal to 1 if user u is served by technology t and is 0 otherwise);
4. flow variables $\phi^f_{ij},\ \forall (i,j) \in A^{\text{R-C}},\ f \in F$ that represent the amount of flow sent on a root or core arc (i,j) for facility f and are introduced to model the connectivity among facilities and central offices in the root-core network.

The MILP problem for 3-ConFL, that we denote as 3-ConFL-MILP, is described by (4)–(12).

In 3-ConFL-MILP, the objective function aims at minimizing the total cost, expressed as the sum of the cost of activating root and core arcs (note that the corresponding summation includes the cost of activated central offices, opened facilities and of activated assignment arcs). The constraints (4) impose that each facility is opened using a single technology,

whereas constraints (5) impose that if a user u is served by technology t, exactly one of the assignment arcs coming from a facility that can serve u is activated on technology t. The constraints (6) link the opening of a facility f on technology t to the activation of assignment arcs involving f and t. The constraints (7) impose the coverage requirement for each technology (we remark that here the weighted sum of users getting a better technology like fiber contributes to satisfying the requirement for the coverage of worse technology like copper). The constraints (8) and (9) jointly model the fiber connectivity within the core network as a multicommodity flow problem that includes one commodity per facility. Specifically, the constraints (8) represent flow conservation in root and core nodes, while (9) are variable upper bound constraints that express the linking between the activation of a root or core arc and the activation of the arc. The constraints (10) impose to activate each wireless facility on at most one power level. Finally, (11) are power-indexed SIR constraints and (12) link the power emission variables to the opening of a wireless facility.

We can obtain a *tighter formulation* of 3-ConFL-MILP using a reformulation of the SIR constraints (11) and (10), which exploit the binary power variables φ_{fl} to derive a special family of power-indexed valid inequalities. These valid inequalities were introduced in [6], as a peculiar family of lifted Generalized Upper Bound (GUB) cover inequalities, and we refer the reader to that paper for an exhaustive description of them. In our case, for a given user u, a serving wireless facility f and a subset of interfering wireless facilities K, these inequalities identify joint power configurations of serving and interfering facilities that deny the service coverage of u and thus correspond with violated SIR constraints. Their form is:

$$y^3_{fu} + \sum_{l=1}^{\lambda} \varphi_{fl} + \sum_{k=1}^{|K|} \sum_{l=q_i}^{|L|} \varphi_{kl} \leq |K| + 1 \,, \tag{3}$$

with $u \in U, \quad \lambda \in L, \quad K \subseteq F \backslash \{f\}$, $(q_1, \ldots, q_{|K|}) \in L^I(t, \Delta, \lambda, \Gamma)$, with $L^I(u, f, \lambda, K) \subseteq L^{|K|}$ representing the subset of interfering power levels of facilities in K that deny the service coverage of u provided by wireless facility f, emitting with power level λ. Such inequalities can be separated and added at the root node to obtain a remarkable strengthening of the linear relaxation of the 3-ConFL-MILP. We denote by *Strong-3-ConFL-MILP*, the problem 3-ConFL-MILP strengthened by inequalities (3).

$$\min \sum_{(i,j) \in A^{\text{R-C}}} c_{ij}\, x_{ij} + \sum_{f \in F} \sum_{t \in T} c^t_f\, z^t_f + \sum_{u \in U} \sum_{t \in T} \sum_{f \in F^t_u} c^t_{fu}\, y^t_{fu} \qquad \text{(3-ConFL-MILP)}$$

$$\sum_{t \in T} z^t_f \leq 1 \qquad f \in F \tag{4}$$

$$\sum_{f \in F^t_u} y^t_{fu} = \; v^t_u \qquad u \in U, t \in T \tag{5}$$

$$y^t_{fu} \leq z^t_f \qquad u \in U, f \in F, t \in T \tag{6}$$

$$\sum_{u \in U} \sum_{\tau=1}^{t} w_u\; v^\tau_u \geq W_t \qquad t \in T \tag{7}$$

$$\sum_{(j,i) \in A^{\text{R-C}}} \phi^f_{ji} - \sum_{(i,j) \in A^{\text{R-C}}} \phi^f_{ij} = \begin{cases} -\sum_{t \in T} z^t_f & \text{if } i = r \\ 0 & \text{if } i \neq r, f \\ +\sum_{t \in T} z^t_f & \text{if } i = f \end{cases} \qquad i \in V^{\text{C}} \cup \{r\}, f \in F \tag{8}$$

$$0 \leq \phi^f_{ij} \leq x_{ij} \qquad (i,j) \in A^{\text{R-C}}, f \in F \tag{9}$$

$$\sum_{l \in L} \varphi_{fl} \leq 1 \qquad f \in F \tag{10}$$

$$a_{fu}\left(\sum_{l\in L} P_l\ \varphi_{fl}\right) - \delta \sum_{k\in F\setminus\{f\}} a_{ku}\left(\sum_{l\in L} P_l\ \varphi_{kl}\right) +$$

$$+ M(1-y^3_{fu}) \geq \delta\ N \qquad f\in F, u\in U \qquad (11)$$

$$\varphi_{fl} \leq y^3_{fu} \qquad f\in F, u\in U, l\in L \qquad (12)$$

$$v^t_u,\ z^t_f,\ x_{ij},\ y^t_{fu},\ \varphi_{fl} \in \{0,1\} \qquad (i,j)\in A, u\in U, f\in F, t\in T, l\in L$$

3 A fast heuristic for solving the 3-ConFL-MILP

The 3-ConFL-MILP can in principle be solved by a state-of-the-art MIP solver, such as IBM ILOG CPLEX [2]. However, the introduction of the SIR constraints (11) make 3-ConFL-MILP a very challenging generalization of the ConFL: we experienced that in the case of realistic instances CPLEX has big difficulties in finding feasible solutions of good quality even after hours of computations. In order to tackle these computational difficulties, we propose to solve the problem by a heuristic that mixes a *probabilistic variable fixing procedure*, guided by the information retrieved by solving (tighter) linear relaxations of 3-ConFL-MILP, with an MIP heuristic based on the execution of an *exact very large neighborhood search*. Our heuristic, formalized in Algorithm 1, is based on considerations about the use of linear relaxations in a variable fixing procedure that have been first made in [3], the paper to which we refer for a more detailed discussion about the mechanisms and features of the heuristic concisely presented here. By solving (tight) linear relaxations, we are able to derive dual bounds for the problem that we can use to compute an *optimality gap* measuring how far the best solution returned from our heuristic is from the best lower bound given by Strong-3-ConFL-MILP. In order to explain how we construct a feasible solution for 3-ConFL-MILP, we first introduce the concept of *facility opening state*:

▶ **Definition 1.** Facility Opening State (FOS): an FOS specifies an opening of a subset of facilities $\bar{F} \subseteq F$ on some technologies such that no facility is open with more than one technology. Formally: $FOS \subseteq F \times T :\ \nexists (f_1,t_1),(f_2,t_2) \in FOS :\ f_1 = f_2 \wedge t_1 \neq t_2$.

Given a FOS and a facility-technology couple $(f,t) \in FOS$, we denote by W^{POT}_{ft} the total weight of users that can be potentially served by f using technology t, i.e. $W^{\text{POT}}_{ft} = \sum_{u\in U^t_f} w_u$. Using this measure, we say that a FOS is *partial for technology* t when the total weight of potential users that can be served by facilities appearing in the FOS using technology t does not reach the minimum coverage requirements W_t for t, i.e. $\sum_{f\in F:(f,t)\in FOS} W^{\text{POT}}_{ft} < W_t$. We also say that a FOS is *complete for technology* t when the total weight is not lower than W_t. Additionally, we call *fully complete* a FOS that is complete for all technologies $t \in T$. We use the completeness concepts to guide the probabilistic fixing of facility opening variables during the construction phase of feasible solutions.

Given a *partial* FOS for technology t, the probability p^{FOS}_{ft} of operating an additional fixing $(f,t) \notin FOS$, thus making a further step towards reaching a complete FOS, is set according to the formula:

$$p^{\text{FOS}}_{ft} = \frac{\alpha\ \tau_{ft} + (1-\alpha)\ \eta_{ft}}{\sum_{(k,t)\notin FOS} \alpha\ \tau_{kt} + (1-\alpha)\ \eta_{kt}}\ , \qquad (13)$$

which convexly combines through factor $\alpha \in [0,1]$ two measures: τ_{ft}, measuring a-priori the attractiveness of operating a variable fixing, and η_{ft}, measuring a-posteriori the attractiveness

of operating a fixing (see [4] for more details). In our case, we set τ_{ft} equal to the optimal value of the linear relaxation Strong-3-ConFL-MILP including the additional fixing $z_f^t = 1$, whereas η_{ft} is equal to the optimal value of the linear relaxation of 3-ConFL-MILP, obtained for a partial fixing of the facility opening variables z.

At the end of a solution construction phase, which is aimed at constructing Σ feasible solutions, the a-priori fixing measures τ are updated, evaluating how good were the fixings made in built solutions. The formula that we use for updates is based on the concept of *optimality gap* (*Gap*) (for a feasible solution of value v and a lower bound L that is available on the optimal value v^* of the problem, we set $Gap(v, L) = (v - L)/v$) and is:

$$\tau_{ft}(h) = \tau_{ft}(h-1) + \sum_{\sigma=1}^{\Sigma} \Delta\tau_{ft}^{\sigma} \text{ with } \Delta\tau_{ft}^{\sigma} = \tau_{ft}(0) \cdot \left(\frac{Gap(\bar{v}, L) - Gap(v_\sigma, L)}{Gap(\bar{v}, L)} \right) \tag{14}$$

where $\tau_{ft}(h)$ is the a-priori attractiveness of fixing (f, t) at fixing iteration h, v_σ is the value of the σ-th feasible solution built in the last construction phase and $\bar{v}$ is the (moving) average of the values of the Σ solutions produced in the previous construction phase. $\Delta\tau_{ft}^{\sigma}$ is a penalization/reward factor for a fixing and depends upon the initialization value $\tau_{ft}(0)$ of τ, combined with the relative variation in the optimality gap that v_σ implies with respect to $\bar{v}$.

Once a *fully complete FOS* is built, we have characterized an opening of facilities that can *potentially* satisfy the requirements on the weighted coverage for each technology. We say "potentially" because the activation of facilities specified by the FOS may not have a feasible completion in terms of connectivity variables and assignment of users of facilities: it is indeed possible that not all the SIR constraints (11) activated by the probabilistic fixing procedure can be satisfied together because of interference phenomena. As a consequence, a complete FOS may be infeasible. To tackle this risk of infeasibility, after the construction of a complete FOS, we execute a *check-and-repair phase*, in which the feasibility of the FOS is checked and, if not verified, we make an attempt to repair and make it feasible. The reparation attempt is based on the same MIP heuristic that we introduce below, with the name MOD-RINS, and that we adopt at the end of the construction phase to possibly improve a feasible solution.

Given a FOS that is complete for all technologies, we check its feasibility and try to find a feasible solution for the complete problem 3-ConFL-MILP by defining a restricted version of 3-ConFL-MILP, where we set $z_f^t = 1$ if $(f, t) \in FOS$. We solve this restricted problem by the MIP solver running with a time limit: if this problem is recognized as infeasible by the solver, we execute the MIP heuristic for reparation. Otherwise, we run the solver to possibly find a solution that is better than the best incumbent solution.

To try to improve an incumbent feasible solution or to repair an infeasible partial fixing of the variables z induced by a complete FOS, we rely on an MIP heuristic that operates a very large neighborhood search *exactly*, by formulating the search as an MILP problem solved through an MIP solver. Specifically, as we did in [3], we rely on a modified version of the RINS Heuristic [8], denoted by MOD-RINS, where the neighborhood is defined combining information from the linear relaxation of 3-ConFL-MILP with that of the current incumbent solution. Due to lack of space, we refer the reader to [3] for a description of the modified RINS algorithm that we have adopted.

The complete algorithm that we used for solving the 3-ConFL-MILP is shown in Algorithm 1. It is based on two nested loops: the outer loop runs until a global time limit is reached; the inner loop is aimed at building Σ feasible solutions, by first defining complete FOSs and then executing the modified heuristic to repair or complete the fixing associated with the FOS. We denote by X^* and X^B the best solutions found by the algorithm in the outer and in the inner loop, respectively. Each run of the inner loop provides for building a complete

Algorithm 1 Heuristic for 3-ConFL-MILP

solve the linear relaxation of Strong-3-ConFL-MILP for every single fixing $z_f^t = 1$ and initialize the values $\tau_{ft}(0)$ with the corresponding optimal values
while a global time limit is not reached **do**
  for $\sigma := 1$ to Σ **do**
    build a complete FOS
    solve 3-ConFL-MILP imposing the fixing $\bar{z}$ specified by the FOS
    if 3-ConFL-MILP with fixing $\bar{z}$ is infeasible **then**
      run MOD-RINS for repairing the fixing $\bar{z}$
    end if
    if a feasible solution $\bar{X}$ is found by the MIP solver and $c(\bar{X}) < c(X^B)$ **then**
      update the best solution found in the inner loop $X^B := \bar{X}$
    end if
  end for
  update τ according to (14)
  if $c(X^B) < c(X^*)$ **then**
    update the best solution found $X^* := X^B$
  end if
end while
run MOD-RINS for improving X^*
return X^*

FOS by considering, in order, fiber, copper and wireless technologies. The complete FOS is built according to the procedure using the probability measures (13) and update formulas (14). The complete FOS provides a (partial) fixing of the facility opening variables $\bar{z}$ and the MIP solver uses it as a basis for finding a complete feasible solution X^* to the problem. If $\bar{z}$ is recognized as an infeasible fixing by the MIP solver, then we run MOD-RINS to try to find a repaired solution. Otherwise, if $\bar{z}$ is feasible and gives rise to a feasible solution that is better than X^B in the current execution of the inner loop, then X^B is updated and the inner loop is iterated. At the end of each run of the inner loop, the a-priori measures τ are updated according to (14) and the best solution X^* is updated, if necessary. When the global time limit is reached, MOD-RINS is applied to X^* in an attempt to improve it.

4 Preliminary computational results

The algorithm was tested on 15 realistic networks instances, which refer to a urban district of the Italian city of Rome that has been discretized into a raster of about 450 elementary small-sized areas. The code was written in C/C++ using IBM ILOG CPLEX Concert Technology. The experiments were performed on a 2.70 GHz Windows machine with 8 GB of RAM and using CPLEX 12.5 as MIP solver. The experiments were run with a time limit of 3600 seconds. The instances consider different traffic generation and user location scenarios and in the considered district 5 central offices and 30 facilities are supposed to be available for deployment. Concerning the setting of the parameters of the heuristic, on the basis of past experience and preliminary tests, we imposed: $\alpha = 0.5$, $\Sigma = 5$, a time limit of 3000 seconds for the execution of the outer loop of Alg. 1 and a limit of 600 seconds for the final execution of the improvement heuristic MOD-RINS. The computational results are presented in Table 1: here, for each instance, we report its ID, the best percentage optimality gap *Gap-CPLEX%* reached by CPLEX within the time limit, the best percentage optimality gap *Gap-Heu%* reached by our heuristic within the time limit. In the case of the heuristic, we note that the gap is obtained combining the best feasible solution found by Algorithm 1 with the best known lower bound obtained by CPLEX using the strengthened formulation Strong-3-ConFL-MILP. The power-indexed version of 3-ConFL-MILP appears to

Table 1 Experimental results

ID	Gap-CPLEX%	Gap-Heu%	ΔGap%
I1	126.69	102.35	-19.21
I2	118.13	82.55	-30.11
I3	136.44	104.94	-23.08
I4	178.11	130.37	-26.80
I5	125.76	86.88	-30.91
I6	109.38	75.00	-31.43
I7	121.66	67.16	-44.79
I8	103.21	57.73	-44.06
I9	163.42	129.01	-21.00
I10	156.29	115.15	-26.32
I11	105.58	82.62	-21.74
I12	101.86	68.98	-32.27
I13	132.21	94.09	-28.83
I14	134.28	89.32	-33.48
I15	123.57	84.87	-31.31

be very difficult to solve for a state-of-the-art solver like CPLEX and the best optimality gap obtained for all instances is (well) over 100%. We believe that this is due to combining two distinct network design problems, wired and wireless, which are challenging already when taken separately. In contrast, our heuristic presents a very good performance, granting an average reduction of 29% in the optimality gaps (the best reduction reaches 44%). This is a very promising outcome and as future work we plan to better investigate the mechanisms of the heuristic and its integration within a branch-and-cut algorithm, which could also more effectively exploit the strength of power-indexed lifted GUB cover inequalities.

References

1 Ahuja, R. K., Magnanti, T., Orlin, J.: Network Flows: Theory, Algorithms, and Applications. Prentice Hall, Upper Saddle River (1993).

2 IBM ILOG CPLEX, `http://www-01.ibm.com/software/integration/optimization`.

3 D'Andreagiovanni, F., Krolikowski, J., Pulaj, J.: A fast hybrid primal heuristic for Multiband Robust Capacitated Network Design with Multiple Time Periods. Applied Soft Computing 26, pp. 497–507 (2015). 10.1016/j.asoc.2014.10.016.

4 D'Andreagiovanni, F., Mett, F., Pulaj, J.: An (MI)LP-based Primal Heuristic for 3-Architecture Connected Facility Location in Urban Access Network Design. Applications of Evolutionary Computation. LNCS 9597, pp. 283–298. Springer, Heidelberg (2016)

5 D'Andreagiovanni, F., Mannino, C., Sassano, A.: Negative Cycle Separation in Wireless Network Design. Network Optimization. LNCS 6701, 51–56. Springer, Heidelberg (2011).

6 D'Andreagiovanni, F., Mannino, C., Sassano, A.: GUB Covers and Power-Indexed Formulations for Wireless Network Design. Management Science 59 (1), pp. 142–156 (2013).

7 D'Andreagiovanni, F.: Revisiting wireless network jamming by SIR-based considerations and multiband robust optimization. Optimization Letters 9(8):1495–1510 (2015).

8 Danna, E., Rothberg, E., Le Pape, C.: Exploring relaxation induced neighborhoods to improve MIP solutions. Math. Program. 102, pp. 71–90 (2005).

9 Dely, P., D'Andreagiovanni, F., Kassler, A.: Fair optimization of mesh-connected WLAN hotspots. Wireless Communications and Mobile Computing 15 (5), pp. 924–946 (2015).

10 Ghazisaidi, N., Maier, M., Assi, C. M.: Fiber-wireless (FiWi) access networks: A survey. IEEE Comm. Mag. 47 (2), pp. 160–167 (2009).

11 Grötschel, M., Raack, C., Werner A.: Towards optimizing the deployment of optical access networks. EURO J. Comp. Opt. 2 (1):17–53 (2014).

12 Gollowitzer, S., Ljubic, I.: MIP models for connected facility location: A theoretical and computational study. Computers & OR 38 (2):435–449 (2011).

13 Gupta, A., Kleinberg, J., Kumar, A., Rastogi, R., Yener, B.: Provisioning a virtual private network: a network design problem for multicommodity flow. Proc. STOC'01, pp. 389–398, ACM, New York (2001)

14 Leitner, M., Ljubic, I., Sinnl, M., Werner, A.: On the Two-Architecture Connected Facility Location Problem. Electronic Notes in Discrete Math. 41, pp. 359–366 (2013).

15 Rappaport, T.S.: Wireless Communications: Principles and Practice, 2nd Edition. Prentice Hall, Upper Saddle River, USA (2001).

Portfolio Optimisation Using Risky Assets with Options as Derivative Insurance*

Mohd A. Maasar[1], Diana Roman[2], and Paresh Date[3]

1 **Department of Mathematics, College of Engineering Design and Physical Science, Brunel University London, UB83PH, U.K.**
mohd.maasar@brunel.ac.uk
2 **Department of Mathematics, College of Engineering Design and Physical Science, Brunel University London, UB83PH, U.K.**
diana.roman@brunel.ac.uk
3 **Department of Mathematics, College of Engineering Design and Physical Science, Brunel University London, UB83PH, U.K.**

Abstract

We introduce options on FTSE100 index in portfolio optimisation with shares in which conditional value at risk (CVaR) is minimised. The option considered here is the one that follows FTSE100 Index Option standards. Price of options are calculated under the risk neutral valuation. The efficient portfolio composed under this addition of options shows that put option will be selected as part of the investment for every level of targeted returns. Main finding shows that the use of options does indeed decrease downside risk, and leads to better in-sample portfolio performance. Out-of-sample and back-testing also shows better performance of CVaR efficient portfolios in which index options are included. All models are coded using AMPL and the results are analysed using Microsoft Excel. Data used in this study are obtained from Datastream. We conclude that adding a put index option in addition to stocks, in order to actively create a portfolio, can substantially reduce the risk at a relatively low cost. Further research work will consider the case when short positions are considered, including writing call options.

1998 ACM Subject Classification G.1.6. Optimization

Keywords and phrases Portfolio optimisation, portfolio insurance, option pricing, mean-CVaR

Digital Object Identifier 10.4230/OASIcs.SCOR.2016.9

1 Introduction

The portfolio selection problem is about how to divide an investor's wealth amongst a set of available securities. One basic principle in finance is that, due to the lack of perfect information about future asset returns, financial decisions are made in the face of trade-offs. Markowitz[8] identified the trade-off faced by the investors as risk versus expected return and proposed variance as a measure of risk. He introduced the concepts of efficient portfolio and efficient frontier and proposed a computational method for finding efficient portfolios.

Following notations given by Roman et al. [15], we consider an example of portfolio selection with one investment period. A rational investor is interested in investing their capital in each of number of available assets so that at the end of the investment period the return is maximised. Consider a set of n assets, with asset $j \in \{1 \dots n\}$ having a return of R_j at the end of the investment period. Since the future price of the asset is unknown,

* This work was partially supported by the Department of Mathematics, Brunel University.

R_j is a random variable. Let x_j be the percentage of capital invested in asset j and let $x = (x_1, \ldots, x_n)$ denote the portfolio choice. This portfolio's return is given as

$$R_x = x_1 R_1 + \ldots + x_n R_n$$

with distribution function $F(r) = P(R_x \leq r)$ depending on the choice of $x = (x_1, \ldots, x_n)$.

The weighs $(x_1, \ldots, x_n)$ belongs to a set of decision vectors given as

$$X = \{(x_1, \ldots, x_n) | \sum_{j=1}^{n} x_j = 1, x_j \geq 0, \forall j \in \{1, \ldots, n\}\} \quad (1)$$

This is the simplest way to represent a feasible set by the requirement that the weights must sum to 1 and no short selling is allowed.

In the mean-risk paradigm, a random variable R_x representing the return of a portfolio x is characterised using two statistics of its distribution: the expected value / mean (large value are desired) and a "risk" value (low values are desired).

An efficient portfolio is the one that has the lowest risk for a specified level of expected return. Varying the level of expected return, we obtain different efficient portfolios [1]. An efficient portfolio is found by solving an optimisation problem in which we minimise risk subject to a constraint on the expected return.

Markowitz [8] proposed variance as a measure of risk. Criticism of variance, mainly due to its symmetric nature that penalised upside potential as well as downside deviations, led to proposal of other risk measures.

The first objective of this paper is to present the results of an empirical study on the effect of different risk measures on portfolio choice. We examine portfolios with the same expected return, obtained by minimising various risk measures: variance together with downside risk measures (lower partial moment (LPM)) and quantile based risk measures (conditional Value-at-Risk (CVaR)).

The second objective of this paper is to analyse the effect of introducing index options, in addition to stocks, to the portfolio optimisation; this is done in the context of mean-CVaR optimisation. Naturally, options are used as insurance against unfavourable outcomes hence they reduce downside risk; this comes however at a cost that reduces overall the return of the portfolio.

The rest of this paper is organised as follows. Risk measures are presented in Section 2. The algebraic formulations of the corresponding mean-risk optimisation models is presented in Section 3. Section 4 describes the background for incorporating an index option in the portfolio optimisation. Computational results are presented in Section 5. Conclusions are drawn in Section 6.

2 Risk Measures

We give a review of mean risk models and risk measures presented by Roman and Mitra in [15] . Risk measures are classified following [1][14][9] into two categories. The first category measures the deviation from a target and concerned with the whole distribution of outcomes. The second category concerns only with the left tail of a distribution.

Adopting the terminology used in [3], *risk measures of the first kind* measure the magnitude of deviations from a specific point. These risk measures can be further divided into symmetric

[1] This corresponds to examples of low mean-low risk trade-offs up to high mean – high risk trade-offs.

risk measures and asymmetric risk measures. Symmetric risk measures are calculated in terms of dispersion of results around a pre-specified target. Namely, symmetric risk measures such as variance and mean absolute deviation (MAD) are mathematically convenient while being not so good in representing the reality. Asymmetric risk measures in the other hand quantify risk based on the observation that an investor's true risk is the downside risk. Lower partial moments and central semi-deviations are among the important asymmetric risk measures.

Risk measures of the second kind measure the overall significance of possible losses. These risk measures are concerned only with a certain number of worst outcomes (the left tail), of the distribution. The commonly used risk measures in this category are Value-at-Risk (VaR) and Conditional Value-at-Risk (CVaR). This section gives brief overview on risk measures (variance, LPM0, VaR and CVaR) that will be used in this work, and review the concept of coherent risk measures.

Variance. Variance is a well-known indicator used in statistics for the spread around the mean of a random variable . The variance of a random variable R_x is defined as its second central moment, the expected value of the square of the deviations of R_x from its own mean;

$$\sigma^2(R_x) = E[R_x - E(R_x))^2]$$

where $E(R_x)$ is the expected value of R_x. The variance of a linear combination of random variables is given as;

$$\sigma^2(aR_1 + bR_2) = a^2\sigma^2(R_1) + b^2\sigma^2(R_2) + 2abCov(R_1, R_2)$$

where R_1, R_2 are random variables, $a, b \in \mathbb{R}$, and $Cov(R_1, R_2)$ is the covariance of R_1 and R_2:

$$Cov(R_1, R_2) = \sigma_{jk} = E[(R_1 - E(R_1))(R_2 - E(R_2))]$$

In portfolio optimisation problems, this property is useful to express the variance of the portfolio return $R_x = x_1R_1 + \ldots + x_nR_n$, as a result from choice $x = (x_1, \ldots, x_n)$ as:

$$\sigma^2(R_x) = \sum_{j=1}^{n}\sum_{k=1}^{n} x_j x_k \sigma_{jk}$$

Thus, variance is expressed as a quadratic function of $x_1, \ldots, x_n$ [8].

Lower Partial Moments (LPM). An asset pricing model using a mean-LPM was first developed by Bawa and Lindenberg [3] and Fishburn [4] in 1977. LPM is a generic name for asymmetric measures that consider a fixed target below which an investor does not want the return to fall. Asymmetric measures provide a more intuitive representation of risk, since upside deviations are not penalised. LPM measures the expected value of deviation below a fixed target value τ.

By letting τ be a predefined target value for the portfolio return R_x, and let $\alpha \geq 0$, the LPM of order α around τ of the random variableR_x with distribution function F is defined as [4]:

$$LPM_\alpha(\tau, R_x) = E\{[max(0, \tau - R_x)]^\alpha\} = \int_{-\infty}^{\tau} (\tau - r)^\alpha dF(r)\,.$$

While τ is a target fixed by decision maker (DM), α is a parameter describing the investor's risk aversion. The larger the α , the more risk-averse is the investor. A decision maker is willing to take a risk in order to minimise the chance that the return falls below τ, provided that the main concern is the failure to meet the target return. For this case, choosing a small α is appropriate. Instead, if small deviations below the target are reasonably harmless when compare to large deviations, the DM prefers to fall lower of τ by small amount. In this case, a larger α is obtained [4].

Value-at-Risk (VaR). One of the most popular quantile-based risk measures is the Value-at-Risk (VaR) [6]. The VaR at confidence level $\alpha \in (0, 1)$ in is defined as the $(1 - \alpha)$-percentile of the portfolio loss distribution, where α is typically chosen as 0.01 or 0.05. Thus, in the calculation of Value-at-Risk at level α of random variable R_x, we suggest that with probability of at least $(1 - \alpha)$, the loss [1] will not exceed VaR. Following definitions presented in [15], the VaR at level α of R_x is defined using the notion of $\alpha-$quantiles:

▶ **Definition 1.** An $\alpha - quantile$ of R_x is a real number r such that

$$P(R_x < r) \leq \alpha \leq P(R_x \leq r)\,.$$

▶ **Definition 2.** The *lower* $\alpha - quantile$ of R_x, denoted by $q_\alpha(R_x)$ is defined as

$$q_\alpha(R_x) = \inf\{r \in \mathbb{R} : F(r) = P(R_x \leq r) \geq \alpha\}\,.$$

▶ **Definition 3.** The *upper* $\alpha - quantile$ of R_x, denoted by $q^\alpha(R_x)$ is defined as

$$q^\alpha(R_x) = \inf\{r \in \mathbb{R} : F(r) = P(R_x \leq r) > \alpha\}\,.$$

▶ **Definition 4.** The *Value-at-Risk at level* α *of* R_x is defined as the negative of the upper $\alpha-$quantile of $R_x : VaR_\alpha(R_x) = -q^\alpha(R_x)$.

The minus sign in the definition of VaR is because $q^\alpha(R_x)$ is likely to be negative. Absolute values are considered in reporting this value in term of "loss".

In general, VaR is not a sub additive measure of risk. It means that the risk of a portfolio can be larger than the sum of the individual risks of its components when measured by VaR (see [16] for examples). Furthermore, VaR is difficult to optimize with standard available methods because VaR is not convex with respect to choice of R_x. This is explained in [7] [10] and references therein. Convexity is an important property in optimization because it removes the possibility of a local minimum being different from a global minimum [13].

Conditional Value-at-Risk (CVaR). Conditional Value-at-Risk [11] [12] was proposed as an alternative quantile-based risk measure. It has been gaining interest from practitioners and academics due to its desirable computational and theoretical properties. As shown in [15], we consider that CVaR is approximately equal to the average of losses greater than or equal to VaR at the same α.

▶ **Definition 5.** The CVaR at level α of R_x is defined as:

$$CVaR_\alpha(R_x) = -\frac{1}{\alpha}E(R_x 1_{\{R_x \leq q^\alpha(R_x)}\}) - q^\alpha(R_x)[P(R_x \leq q^\alpha(R_x)) - \alpha]\}$$

where

$$1_{Relation} = \begin{cases} 1, & \textit{if} \text{ Relation } \textit{is true}; \\ 0, & \textit{if} \text{ Relation } \textit{is false}. \end{cases}$$

[1] In our context we refer negative returns as positive losses. Therefeore, any loss related to random variable R_x is represented by a random variable $-R_x$

2.1 CVaR calculation and optimisation

The following results is used in CVaR optimisation; It was proven by Rockafellar and Uryasev in [12]. Let R_x be a random variable depending on a decision vector x that belongs to a feasible set X as defined by 1 and let $\alpha \in (0,1)$. CVaR of the random variable R_x for confidence level α is denoted by the $CVaR_\alpha(x)$. Consider the function:

$$F_\alpha(x,v) = \frac{1}{\alpha}E[-R_x + v]^+ - v,$$

$$[u]^+ = \begin{cases} u, & \textit{if } u \geq 0; \\ 0, & \textit{if } u < 0. \end{cases}$$

Then:

1. As a function of u , F_α is finite and continuous (thus convex) and

   $$CVaR_\alpha(x) = \min_{v\in\mathbb{R}} F_\alpha(x,v)\,.$$

   In addition, the set consisting of the values of v for which the minimum is attained, denoted by $A_\alpha(x)$, is a non-empty, closed and bounded interval that contains $-VaR_\alpha(R_x)$. In some cases, the set may be formed by just one point.
2. Minimising $CVaR_\alpha$ with respect to $x \in X$ is equivalent to minimising F_α with respect to $(x,v) \in X \times \mathbb{R}$:

   $$\min_{x\in X} CVaR_\alpha(x) = \min_{(x,v)\in X\times\mathbb{R}} F_\alpha(x,v)\,.$$

   In addition, a pair (x^*, v^*) minimises the right hand side if and only if x^* minimises the left hand side and $v \in A_\alpha(x^*)$.
3. $CVaR_\alpha(x)$ is convex with respect to x and F_α is convex with respect to (x,v).
   Thus, if the set X of feasible decision vectors is convex, minimising CVaR is a convex optimisation problem.
   In practice, a portfolio return R_x is a discrete random variable because the random returns are usually described by their realisations under various scenarios. This simplifies the calculation and optimisation of CVaR as it makes the optimisation problems above as linear programming problems. Suppose that R_x has m possible outcomes $r_{1x}, \ldots, r_{mx}$ with probabilities $p_1, \ldots, p_m$ with $r_{ix} = \sum_{j=1}^{n} x_j r_{ij}, \forall i \in \{1 \ldots m\}$, then:

   $$F_\alpha(x,v) = \frac{1}{\alpha}\sum_{i=1}^{m} p_i[v - r_{ix}]^+ - v = \frac{1}{\alpha}\sum_{i=1}^{m} p_i[v - \sum_{j=1}^{n} x_j r_{ij}]^+ - v\,.$$

   This formulation will be used for the mean-CVaR optimisation model in the next section. In contrast to VaR, the CVaR is a convex function of the portfolio weights $x = (x_1, \ldots, x_n)$. It is obvious that $CVaR_\alpha(x) \geq VaR_\alpha(x)$ for any portfolio $x \in X$. Thus, minimising CVaR can be used to limit the VaR of a portfolio. Furthermore CVaR is known to be a coherent risk measure. We give a review on coherent risk measure in the next subsection.

2.2 Coherent Risk Measures

Consider a set V of random variables representing future returns or net worth of portfolios. The function $\rho : V \mapsto \mathbb{R}$ is said to be a coherent risk measure if it satisfies the following four axioms:

1. **Subadditivity:** For all $v_1, v_2 \in V, \rho(v_1 + v_2) \leq \rho(v_1) + \rho(v_2)$.
2. **Translation Invariance:** For all $v \in V, a \in \mathbb{R}, \rho(v + a) = \rho(v) - a$.
3. **Positive Homogeneity:** For all $v \in V, a \geq 0, \rho(av) = a\rho(v)$.
4. **Monotonicity:** For all $v_1, v_2 \in V$ such that $v_1 \geq v_2, \rho(v_1) \leq \rho(v_2)$.

The four axioms that define coherency were introduced by Artzner et.al [2]. The subadditivity axiom ensures that the risk associated with the sum of two assets cannot be larger than the sum of their individual risk values. This property requires that financial diversification can only reduce risk. Translation invariance means that receiving a sure amount of a reduces the risk quantity by a. Positive homogeneity implies that the risk measure scales proportionally with the size of the investment. Finally, monotonicity implies that when one investment almost surely performs better than another investment, its risk must be smaller.

From all risk measures discussed in this section, only the CVaR is a coherent risk measure. VaR fails to satisfy subadditivity axiom, and the mean-standard deviation risk measure does not satisfy monotonicity axiom.

3 Mean-risk Optimisation Models

In this section we provide the algebraic formulation of mean-risk models that will be used in this work, namely mean-variance, the mean-LPM of target 0 and order 1, and mean-CVaR optimisation modes. At the end of this section we will give a brief equivalence of mean-CVaR optimisation with a model for maximising returns under a CVaR constraint.

3.1 Algebraic form of the mean-risk models

In this subsection we present below the algebraic form of the three mean-risk models used in our computational analysis. We will use the following notation:

Input data

$$\begin{array}{rcl} m & = & \text{the number of (equally probable) scenarios;} \\ n & = & \text{the number of assets;} \\ r_{ij} & = & \text{the return of asset } j \text{ under scenario } i;\ j = 1 \ldots n,\ i = 1 \ldots m; \\ \mu_j & = & \text{the expected rate of return of asset } j;\ j = 1 \ldots n; \\ \sigma_{kj} & = & \text{the covariance between returns of asset } k \text{ and asset } j; k, j = 1 \ldots n; \\ d & = & \text{level of targeted return for the portfolio.} \end{array}$$

The decision variables

$$x_j \quad = \quad \text{the fraction of the portfolio value invested in asset } j, j = 1 \ldots n.$$

3.2 The Mean-Variance Model (MV)

$$\min_x \quad \sum_{j=1}^{n} \sum_{k=1}^{n} \sigma_{kj} x_j x_k$$

subject to:

$$\sum_{j=1}^{n} \mu_j x_j \quad \geq \quad d \quad ; \quad \forall x \in X$$

3.3 The Mean-Expected Downside Risk model (M-LPM0)

For this model, in addition to the decision variables x_j, there are m decision variables, representing the magnitude of negative deviations of the portfolio return from the zero value, for every scenario $i \in \{1 \dots m\}$:

$$y_i = \begin{cases} -\sum_{j=1}^{n} r_{ij} x_j, & \text{if } \sum_{j=1}^{n} r_{ij} x_j \leq 0; \\ 0, & \text{otherwise.} \end{cases}$$

$$\begin{aligned} \min \quad & \frac{1}{m} \sum_{j=1}^{m} y_i \\ \text{subject to:} \quad & \\ & -\sum_{j=1}^{n} r_{ij} x_j \leq y_i \quad ; \quad \forall i \in \{1 \dots m\} \\ & y_i \geq 0 \quad ; \quad \forall i \in \{1 \dots m\} \\ & \sum_{j=1}^{n} \mu_j x_j \geq d \quad ; \quad \forall x \in X \end{aligned}$$

3.4 The Mean-CVaR$_\alpha$ Model (M-CVaR$_\alpha$)

For this model, in addition to the decision variables x_j, there are $m+1$ decision variables. The variable v represents the negative of an α-quantile of the portfolio return distribution. Thus, when solving this model, the optimal value of the variable v may be used as an approximation for VaR$_\alpha$. The other m decision variables represent the magnitude of negative deviations of the portfolio return from the α-quantile, for every scenario $i \in \{1 \dots m\}$:

$$y_i = \begin{cases} -v - \sum_{j=1}^{n} r_{ij} x_j, & \text{if } \sum_{j=1}^{n} r_{ij} x_j \leq -v; \\ 0, & \text{otherwise.} \end{cases}$$

$$\begin{aligned} \min \quad & v + \frac{1}{\alpha m} \sum_{j=1}^{m} y_i \\ \text{subject to:} \quad & \\ & \sum_{j=1}^{n} -r_{ij} x_j - v \leq y_i \quad ; \quad \forall i \in \{1 \dots m\} \\ & y_i \geq 0 \quad ; \quad \forall i \in \{1 \dots m\} \\ & \sum_{j=1}^{n} \mu_j x_j \geq d \quad ; \quad \forall x \in X \end{aligned}$$

4 Mechanism of Option Pricing and Portfolio Management

An option is a financial derivative described as a contract when the holder of the contract is given a right to exercise a deal, but the holder is not obliged to exercise this right. Financial options are traded both on exchanges and in the over-the-counter market.

There are two basic types of options namely calls and puts. A call option gives the holder the right to buy (not the obligation) the underlying asset (stock, real estate, etc.) at a

certain price at a specified period of time. A put option gives the holder the right to sell the underlying asset at a certain price at a specified period of time. The price of underlying stated in the contract is known as the exercise price or strike price while the date in the contract is known as the expiration date or maturity [5].

There are four types of participant in option markets, *buyers of calls*, *sellers of calls*, *buyers of puts*, and *seller of puts*. Most common options that are being exercised today are either American options or European options, which differ in period of exercising the option. American options can be exercised at any time from the date of writing up to the expiration date, while European options can only be exercised on the expiration date.

4.1 Value of options

The value of an option can be broken into two components called intrinsic value and time value. *Intrinsic value* is the difference between the value of the underlying and the exercise price. The intrinsic value of an option may be either positive or zero, but it can never be negative. This is because the contract involves no liability on the part of the option holder, where the option holder can walk away without exercising the option.

▶ **Example 6.** For a holder of call option, if the value of underlying (example: stock) is less than the option exercise price, the option is referred as being *out of the money* (OTM). If the value of stock is greater than the exercise price, the option is referred to as being *in the money* (ITM). If the value of the stock is equal to the exercise price, the option is referred to as being at the money (ATM).

Mathematically, value of an option is represented in term of option payoff function. An option payoff function, evaluated as a function of the underlying stock price S_T , at maturity. Consider put and call options with strike price K, the payoff function is given as:

$$V_{put}(S_T) = \max\{0, K - S_T\}, \text{ and } V_{call}(S_T) = \max\{0, S_T - K\},$$

respectively.

▶ **Example 7.** Assume that an investor is holding a portfolio consisting of a stock (long) and a put option on the same stock (long) with strike price K. The payoff function of the portfolio,V_{pf}, is therefore given as

$$V_{pf}(S_T) = S_T + V_{put}(S_T) = \max\{K, S_T\}\,.$$

This payoff function shows that the put option with strike price K secure the portfolio value at maturity from dropping below K.

Risk-neutral valuation [5] is used in our implementation towards incorporating option into the mean-risk portfolio optimisation problems. Thus, we assume that in the long-run, the price on an option will equal to the expected future payoff of the option itself, discounted at a risk-free rate r, over maturity period $T - t$. We denote C_t and P_t as our empirical prices for call and put respectively, so

$$P_t \text{ or } C_t = E[\text{payoffs}]e^{-r(T-t)}\,.$$

4.2 Incorporating option into portfolio optimisation

In the case of stocks, obtaining the parameters r_{ij} necessary in order to implement the optimisation models in 4.1. is straightforward, if we use historical data in order to generate

scenarios. In this case, we monitor the prices of the corresponding shares at interval of times equal to the investment period of the portfolio model; we then compute the historical returns and use them as "scenarios", assuming that past will repeat in the future. Alternatively, we can simulate prices for the shares at the end of the investment period, and using the current known prices, to obtain simulations for the returns.

Consider now the case when one of the assets is an option (on a generic underlying asset), with the same maturity as the investment period of the portfolio problem. In this case, obtaining the scenarios for its return at the end of the investment period is different, since the return depends on the price of the underlying asset at expiration.

In this work, we introduce a call and a put option on the FTSE100 index as two additional assets to the existing components of FTSE100.

In this case, we need to simulate the price of FTSE100 at the end of the investment period. We have used historical rates of return for the component assets of FTSE100. To keep consistency, we do the same for the index itself: we monitor the historical rates of return and, using the current (known) price, we simulate prices for FTSE100 at the end of the investment period.

The corresponding pay-offs of the options can then be calculated for any strike price K. The return of the option is calculated then using the current price of the option and the simulated pay-offs. If there is no price available for an option with strike price K and the given maturity, a correct price can be calculated, for example using the arbitrage free / risk neutral pricing using the current risk free rate of return.

To summarise, we use the following method :

1. We compute the historical rates of return for the underlying FTSE 100 for a period of time equal to investment period.
2. The returns from step 1 are used to simulate next period's underlying (FTSE100) value, given its current value:

$$S_{t+1|t} = S_t(1 + r_{t+1})\,.$$

3. Using known strike price for call and put options, K_c and K_p, and one period simulated underlying asset value $S_{(t+1|t)}$, we simulate option payoffs at their maturity $t+1$. Thus the payoffs for call and put are given respectively as:

$$C_{t+1|t} = \max(S_{t+1} - K_c, 0), \text{ and } P_{t+1|t} = \max(K_p - S_{t+1}, 0).$$

4. Using the simulated payoff above, option returns are computed by:

$$r_{t+1|t,c} = \frac{C_{t+1|t}}{C_t} - 1, \text{ and } r_{t+1|t,p} = \frac{P_{t+1|t}}{P_t} - 1$$

5 Computational Results

This section aims to investigate the behaviour of mean-risk models exhibited formulated in Section 3 when used in application of investment portfolio selection. We consider three different mean-risk models, with risk measured by variance (denoted by M-V), the expected downside risk by the lover partial moment of order 1 and target 0 (M-LPM0), and CVaR for 5% confidence level (M-CVaR$_{0.05}$).

We first work with only stocks and we consider the efficient portfolios in the mean-risk models above for several levels or expected return. We investigate their properties, in terms of their composition, the in-sample performance, and the out-of-sample performance. At the

Table 1 The Euclidean distance between efficient portfolios with expected return $d_1 = 0.01$.

Models		Value of D
M-V	M-LPM0	0.245071661
M-V	M-CVaR$_{0.05}$	0.298300895
M-LPM0	M-CVaR$_{0.05}$	0.398379982

same time, the equivalence of M-CVaR$_{0.05}$ and model for maximisation of expected value with CVaR constraint (Max-E) is shown for the same level of alpha.

Finally, we include FTSE 100 index options (put and call), in addition to FTSE100 stocks. We implement the mean-CVaR model with this new universe of assets and compare the performance of the resulting optimal portfolios with that of stocks-only portfolios.

5.1 Data set

The data used for this analysis is drawn from the FTSE100 index. Our investment period is one month. The monthly returns of the 87 stock components of the index from January 2005 until January 2015 were considered. The dataset for the in-sample analysis has 100 time periods from January 2005 until May 2013. For the out-of-sample analysis, the behaviour of the portfolio obtained is examined over the twenty months period of June 2013 until January 2015. The models were implemented in AMPL and solved using CPLEX 12.5 optimisation solver.

5.2 Methodology

The characteristics of efficient portfolios may vary depending on targeted return, d. Based on our data set, the maximum level of asset return is 0.0349 and the minimum is at -0.007323. Thus, for simplicity of implementation and for feasible possible solutions, we chose three different level of d as $d_1 = 0.01$, $d_2 = 0.02$, and $d_3 = 0.03$. We solve the three mean-risk models considered above for every level of expected return d_1, d_2, and d_3. We also give a fair comparison between two equivalent optimisation models: of M-CVaR$_{0.05}$ and Max-E for the same alpha level.

5.3 The composition of the efficient portfolios

The difference of the composition of two portfolios $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ is described by using the Euclidean distance between vectors x and y. This quantity is denoted as $D_{(x,y)} = \sqrt{(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2}$.

It is observed that the composition of the M-CVaR$_{0.05}$ efficient portfolios differs substantially from the composition of the other efficient portfolios. Table 1 presents the values of the indicator D for the efficient portfolios with expected return $d_1 = 0.01$. The M-CVaR$_{0.05}$ efficient portfolios have relatively different composition with the other two mean-risk models. The Euclidean distance between the M-CVaR$_{0.05}$ and the M-LPM0 is 0.298 and the distance with the M-V is 0.398. The same features is also observed for the other two levels of expected return.

We also observe the number of assets that constitute the efficient portfolios. It is seen that the model with the risk measure of the first kind, the variance, produced portfolios with the highest number of component assets, while M-CVaR$_{0.05}$ model produce portfolios with the lowest number of component assets. This is consistent with the modelling standard,

Table 2 The number of assets in the composition of mean-risk portfolios.

	M-V	M-LPM0	M-CVaR$_{0.05}$
$d_1 = 0.01$	18	12	9
$d_2 = 0.02$	13	11	8
$d_3 = 0.03$	6	6	4

Table 3 Statistics for the mean-risk efficient distributions with expected value $d_1 = 0.01$.

	M-V	M-LPM0	M-CVaR$_{0.05}$
Median	0.013642802	**0.015415625**	0.011657133
Standard Deviation	**0.033783896**	0.037980983	0.03830069313
Skewness	-0.572809564	-0.699597028	**0.361368572**
Minimum	-0.091575414	-0.13878571	**-0.075281385**
Maximum	0.096340696	0.115140014	**0.1340699**

since diversification is the basis of the mean-variance portfolio theory. The number of assets composition is shown in Table 2. It is also noticed that as the level of portfolio expected return increases, the number of component assets decreases. This is also consistent with the modelling standard that at lower levels of return, an efficient portfolio has more assets in composition in order to lower the risk.

5.4 In-sample analysis

The return distributions of the efficient portfolios are discrete with 100 equally probable outcomes. We analyse these distributions using in sample parameters of standard deviation, skewness, minimum, maximum, and range. We compare sets of three distributions, each having the expected values of d_1, d_2, and d_3.

For a portfolio distribution, it is desirable to have smaller standard deviation and range, and to have larger median, skewness, minimum, and maximum.

Table 3 presents the results obtained for the level of targeted return $d_1 = 0.01$. The M-V efficient portfolio has the lowest standard deviation. The return distribution of the M-CVaR efficient portfolios achieve the best values for the parameters concerned with the left tail of distributions. It has the highest skewness, highest minimum and also highest maximum. Notice that M-CVaR efficient distribution always positively skewed while the other models, the distribution are negatively skewed. Similar results is obtained for other levels of d as shown in Table 4 and 5.

5.5 Out-of-sample analysis

We analysed the performance of the efficient portfolios over the twenty time periods following the last period of in-sample data. The results of the out-of-sample analysis were consistent with those of the in-sample analysis. The consistency is in the sense of the portfolios selected under M-CVaR models were distinct from the other two mean-risk models. This is shown via Figure 1, where it may be noticed that a generally good performance of M-CVaR models are obtained. The similar figures for different in-sample mean returns also shown in Figure 2 and 3.

Figure 1 exhibits the compounded returns of the efficient portfolios with in-sample mean returns $d_2 = 0.02$. It is seen that the M-CVaR efficient portfolio performs better than the

Table 4 Statistics for the mean-risk efficient distributions with expected value $d_2 = 0.02$.

	M-V	M-LPM0	M-CVaR$_{0.05}$
Median	0.026989273	**0.019547821**	0.027136642
Standard Deviation	**0.045539089**	0.048306959	0.053504299
Skewness	-1.068379489	-0.541590574	**0.106153237**
Minimum	-0.169911236	-0.169067343	**-0.147248853**
Maximum	0.113089218	0.132563297	**0.189886327**

Table 5 Statistics for the mean-risk efficient distributions with expected value $d_3 = 0.03$.

	M-V	M-LPM0	M-CVaR$_{0.05}$
Median	0.034052445	**0.025471156**	0.033057945
Standard Deviation	**0.077244906**	0.080794207	0.084842553
Skewness	-0.577646499	-0.307744989	**0.199328725**
Minimum	-0.254267371	-0.269452992	**-0.213015378**
Maximum	0.186538415	0.200218938	**0.274484094**

other two portfolios. Statistical parameters for out-of-sample results also show that M-CVaR had the highest mean. However, note that the out-of-sample performance for $d_3 = 0.03$ is fairly different from our expected performance. Further investigations are required to explain this behaviour in the near future.

5.6 Introducing index options in the universe of assets

In this subsection, we consider in addition to the stocks described in 5.1, a put and a call option on FTSE 100 with maturity one month and with strike price $K = 6583.1$, which is equal to the current price of FTSE 100. We compute the prices for these options using risk free valuation and simulate the rates of return as described in section 4.3. We test the performance on two mean-risk models, the M-V and the M-CVaR$_{0.05}$, for d_1, d_2, and d_3. Based on portfolio composition, it is interesting that both models include index put option as one of the components in the efficient portfolio for every level of targeted returns, with range of weights from 1.2% to 2.9% of overall wealth. The call option is however not selected at any level of targeted returns. Table 6 summarizes these figures.

Another out-of-sample finding is on the risk measures comparison between portfolio with stock only (S-portfolios) versus portfolio with stocks and options (OS-portfolios). It is observed that for every cases of d, in the case of M-CVaR$_{0.05}$, risk associated with OS-portfolios always lower than of S-portfolios. For example of $d = 0.01$, the average loss in the worst 5% cases is only 2.34% of initial investment of OS-portfolios. At the same level, the average loss fo S-portfolios is nearly doubles at 5.63%. We present the values in Table 7.

We also vary our strike price K to different levels such as at the average index point level and the minimum index point level over the simulated index points. Interestingly, the portfolio composition gives similar results as we present here; (1) It is observed that no call option will be selected to the portfolio in all cases, and (2) the risk measures for OS-portfolios are lower than of S-portfolios at every level of d.

To continue with the implementation of options into our portfolio optimisation problems, a backtesting is conducted to see the accuracy of our portfolio in predicting actual results.

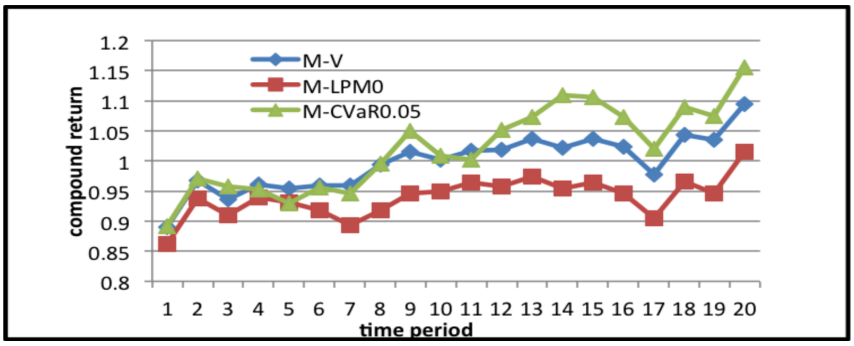

Figure 1 The out-of-sample performance of the mean-risk efficient portfolios with in-sample mean returns $d_2 = 0.02$

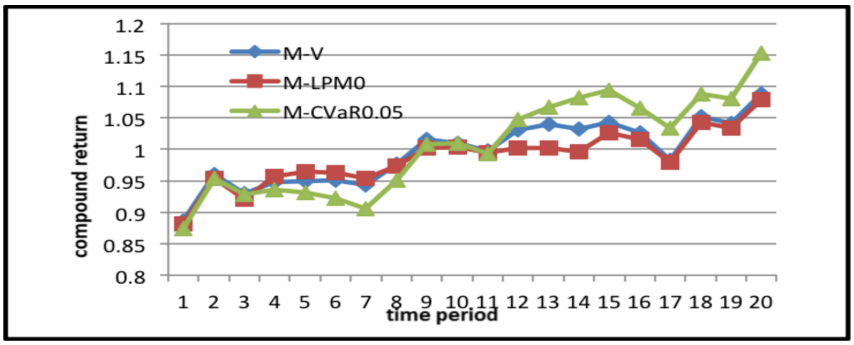

Figure 2 The out-of-sample performance of the mean-risk efficient portfolios with in-sample mean returns $d_1 = 0.01$

5.7 Backtesting

The purpose of backtesting is to see how the optimized portfolios would have performed in reality. Using the same data drawn from FTSE100 as before, we use the optimized weights of each stocks (and options) to evaluate the next period's expected return.

The backtest plan that we use in this study follows the following Figure 4.

We analysed the performance of our strategy by comparing the backtest results of portfolio with stock only (S-portfolios) versus portfolio with stocks and options (OS-portfolios). The backtest is performed over 12-month period (until t=12) by looking at the growth rate of both portfolios. Results show that the growth of OS-portfolios is higher until the 8th month. In these time periods, we observed that options are beneficial to the portfolios, while the S-portfolios suffer from bigger losses or lower returns. For the last 4-month period we can see that the growth of S-portfolios growing stronger. In these periods, it is made clear that there are some scenarios when buying put options is just a cost of insurance for a portfolio against huge losses. Figure 5 exhibit the growth that we mentioned here.

The expected return for each period is shown in the following Table 8 , by comparing the returns for the S-portfolios with the OS-portfolios. It is observed that the maximum return of S-portfolio is 9.3% and the maximum for the OS-portfolio is at 8.1%. Whereas, the minimum for S-portfolio is the loss of 10.9% while the minimum for OS-portfolio is only a loss of 3.1%. We interpret the overall performance of the two portfolios by looking at the average of these expected returns and their standard deviations. It is clearly seen that the average expected returns of S-portfolios is slightly higher of 0.5% against the returns

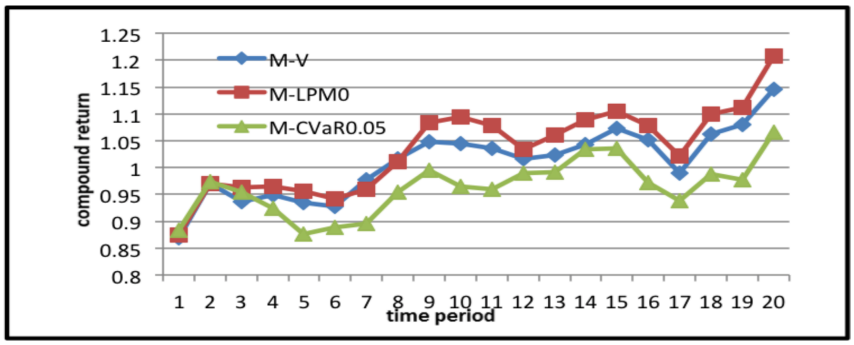

Figure 3 The out-of-sample performance of the mean-risk efficient portfolios with in-sample mean returns $d_3 = 0.03$

Table 6 The number of assets in the composition of mean-risk portfolios with weight of index put option.

	M-V		M-CVaR$_{0.05}$	
	Number of assets	Weight of put	Number of assets	Weight of put
$d_1 = 0.01$	19	0.0124823	17	0.0217767
$d_2 = 0.02$	15	0.0149846	13	0.0235232
$d_3 = 0.03$	6	0.020194	6	0.02948

of OS-portfolios. However, the deviation measure of S-portfolios is s higher than deviation measure of OS-portfolios by 2.4%. These indicate that we have a better performance of CVaR efficient portfolios in which index options are included.

The expected return for each period is shown in the following Table 8 , by comparing the returns for the S-portfolios with the OS-portfolios. It is observed that the maximum return of S-portfolio is 9.3% and the maximum for the OS-portfolio is at 8.1%. Whereas, the minimum for S-portfolio is the loss of 10.9% while the minimum for OS-portfolio is only a loss of 3.1%. We interpret the overall performance of the two portfolios by looking at the average of these expected returns and their standard deviations. It is clearly seen that the average expected returns of S-portfolios is slightly higher of 0.5% against the returns of OS-portfolios. However, the deviation measure of S-portfolios is s higher than deviation measure of OS-portfolios by 2.4%. These indicate that we have a better performance of CVaR efficient portfolios in which index options are included.

6 Conclusion

We consider three mean-risk models, where the risk measures are variance, the lower partial moment of oder 1 and target 0 and CVaR at confidence level 95%. They are conceptually measuring risk very differently, as variance is a symmetric risk measure that penalises deviations from mean on either side of the distribution. Lower partial moments are asymmetric risk measures that penalise only negative deviations from a fixed target, while CVaR is a tail/quantile risk measure that only looks at a pre-specified percentage of worst case losses.

We implement the risk models in AMPL using a dataset drawn from FTSE100 with 87 stocks. For each mean-risk models, we consider the efficient portfolios at expected rate of return 1%, 2% and 3% respectively – corresponding to low mean-low risk, medium mean-medium risk, high mean-high risk trade-offs. We observed that the mean-variance portfolios are the most diversified while the mean-CVaR efficient portfolios the least diversified. The

Table 7 Risk measure quantities for M-CVaR$_{0.05}$ for efficient portfolio with and without options.

	M-CVaR$_{0.05}$	
	S-portfolio	OS-portfolio
$d_1 = 0.01$	5.63%	2.34%
$d_2 = 0.02$	8.05%	2.67%
$d_3 = 0.03$	13.42%	8.54%

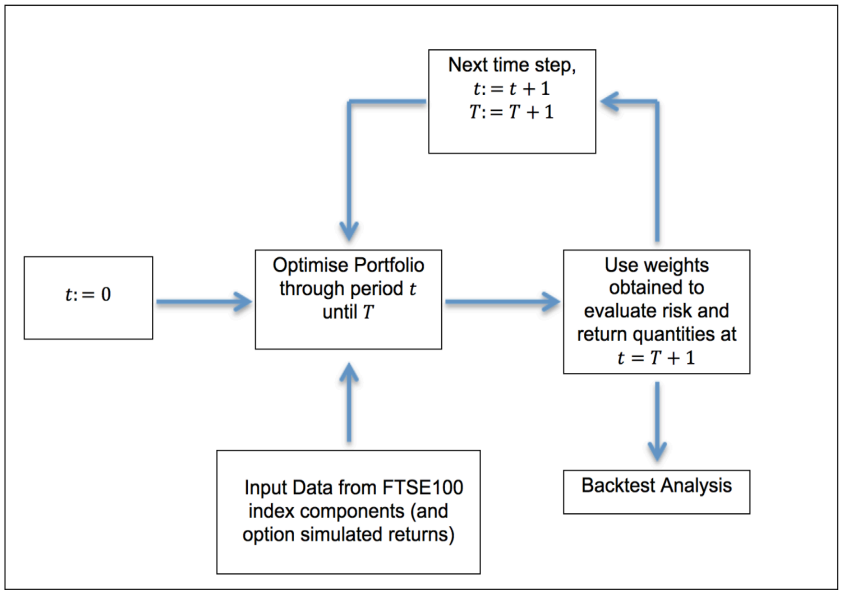

Figure 4 The Backtesting Plan.

in-sample characteristics of their return distributions are somewhat as expected, considering the nature of the risk measures. The mean-CVaR efficient portfolios have the best left tail, with the highest minimum and highest skewness. The out-of-sample performance, evaluated on the next 12 months following the last price observations, show a generally better performance of mean-CVaR efficient portfolios, although this is not consistent across.

We include FTSE100 index options (calls and puts) in the mean-variance and mean-CVaR portfolio optimisation. The maturity of the options is the same with the investment period (one month in our computational work) and we experimented with different strike prices; for each of these we calculated the price of the option under the risk neutral valuation and generated the scenarios for options returns. For each of the strike prices considered, the optimal portfolio weight of the call is zero hence there is no investement in it, while in the majority of cases, the optimal weight of the put option was around 2% of the portfolio value.

This is somewhat expected since the return of the portfolio of stocks, even though actively constructed via risk minimisation is positively correlated with index's return. An investment in a portfolio of stocks from FTSE100 (long positions) is somewhat assuming that the price of FTSE100 is on increase. The put option acts hence as a type of insurance for the cases when the FTSE100 prices are on the decrease.

The computational results are interesting not because the risk (either measured by variance or CVaR) is decreased with the introduction of option; this is natural since the option is simply an additional asset which can only improve or keep the same risk value.

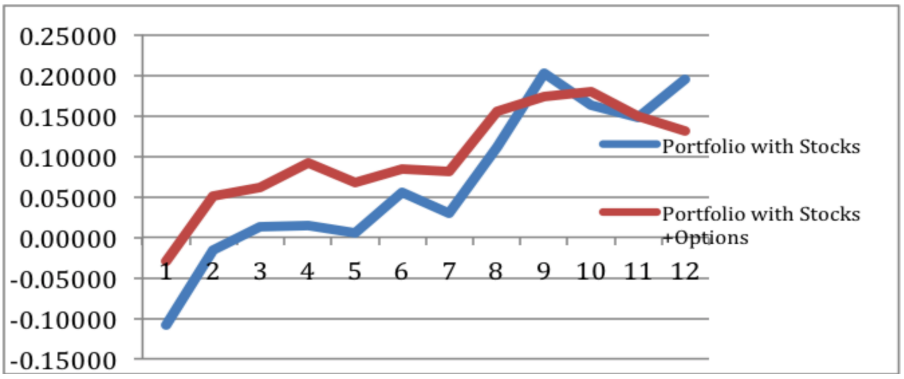

Figure 5 Portfolio Growth Comparison between Stock Portfolio (S-portfolio) and Stock+Options Portfolio (OS-portfolio).

Table 8 Expected Returns for S-portfolios and OS-portfolios.

	Expected Returns	
Backtest Periods	S-portfolio	OS-portfolio
1	**-0.10889**	-0.03016
2	**0.09329**	**0.08102**
3	0.02801	0.01105
4	0.00209	0.03019
5	-0.00969	-0.02504
6	0.05007	0.01787
7	-0.02598	-0.00431
8	0.08269	0.07479
9	0.09147	0.01838
10	-0.03970	0.00561
11	-0.01483	**-0.03051**
12	0.04622	-0.01729
Average	**0.01623**	0.01097
Standard deviation	0.060752874	**0.037197654**

The results are interesting because the decrease in risk is substantial. For example, with the CVaR minimisation at 2% in sample expected rate of return, the optimal CVaR in the case of stocks only is 8.05%, while in the case of stocks + put is 2.67%, for only 2.35% of the portfolio value invested in the put (refer Table 6 and 7).

The backtesting results show that the portfolios composed of stocks and options had substantially different realised returns, compared with the stocks only portfolio. It is somewhat expected that the stocks only portfolio has an average slightly better returns there will be cases when the put has zero pay off and -100% return. However, the realised returns of the portfolios including the option have a better minimum– the lowest realised rate of return is 8%, as compared to -11% in the case of stocks only portfolios – and also much lower standard deviation. As a conclusion, adding a put index option in addition to stocks, in order to actively create a portfolio, can substantially reduce the risk at a relatively low cost. As a future research work, we would like to consider the case when short positions are considered, including writing call options.

References

1. P. Albrecht. *Risk Measures*. Encyclopaedia of Actuarial Science (J. Teugels and B. Sundt, eds). John Wiley and Sons, New York, 2004.
2. Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical finance*, 9(3):203–228, 1999.
3. Vijay S Bawa and Eric B Lindenberg. Capital market equilibrium in a mean-lower partial moment framework. *Journal of Financial Economics*, 5(2):189–200, 1977.
4. Peter C Fishburn. Mean-risk analysis with risk associated with below-target returns. *The American Economic Review*, 67(2):116–126, 1977.
5. John C Hull. *Options, futures, and other derivatives*. Pearson Education India, 2006.
6. Philippe Jorion. *Value at risk: the new benchmark for managing financial risk*, volume 3. McGraw-Hill New York, 2007.
7. Nicklas Larsen, Helmut Mausser, and Stanislav Uryasev. *Algorithms for optimization of value-at-risk*. Springer, 2002.
8. H. Markowitz. Portfolio selection. *Journal of Finance*, 17(1):77–91, 1952.
9. W. Ogryczak and T. Sliwinski. On solving linear programs with ordered weighted averaging objective. *European Journal of Operations Research*, 148:80–91, 2003.
10. Jong-shi Pang and Sven Leyffer. On the global minimization of the value-at-risk. *Optimization Methods and Software*, 19(5):611–631, 2004.
11. R Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.
12. R Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of banking & finance*, 26(7):1443–1471, 2002.
13. Ralph Tyrell Rockafellar. *Convex analysis*. Princeton university press, 2015.
14. R.T. Rockafeller, S. Uryasev, and M. Zabarankin. Deviation measures in risk analysis and optimization. In *Research Report 2002-7, Risk Management and Engineering Lab/Center for Applied Optimization*, University of Florida, 1994.
15. D. Roman and G. Mitra. Portfolio selection models: A review and new directions. *Wilmott Journal*, 1(2):69–85, 2009.
16. Dirk Tasche. Expected shortfall and beyond. *Journal of Banking & Finance*, 26(7):1519–1533, 2002.