AIRO Springer Series 7

Raffaele Cerulli · Mauro Dell'Amico · Francesca Guerriero · Dario Pacciarelli · Antonio Sforza *Editors*

Optimization and Decision Science

ODS, Virtual Conference, November 19, 2020

AIRO
ASSOCIAZIONE ITALIANA DI RICERCA OPERATIVA
OPTIMIZATION AND DECISION SCIENCE

Springer

AIRO Springer Series

Volume 7

The AIRO Springer Series focuses on the relevance of operations research (OR) in the scientific world and in real life applications.

The series publishes peer-reviewed only works, such as contributed volumes, lectures notes, and monographs in English language resulting from workshops, conferences, courses, schools, seminars, and research activities carried out by AIRO, Associazione Italiana di Ricerca Operativa - Optimization and Decision Sciences: http://www.airo.org/index.php/it/.

The books in the series will discuss recent results and analyze new trends focusing on the following areas: Optimization and Operation Research, including Continuous, Discrete and Network Optimization, and related industrial and territorial applications. Interdisciplinary contributions, showing a fruitful collaboration of scientists with researchers from other fields to address complex applications, are welcome.

The series is aimed at providing useful reference material to students, academic and industrial researchers at an international level.

Should an author wish to submit a manuscript, please note that this can be done by directly contacting the series Editorial Board, which is in charge of the peer-review process.

THE SERIES IS INDEXED IN SCOPUS

More information about this series at http://www.springer.com/series/15947

Raffaele Cerulli • Mauro Dell'Amico •
Francesca Guerriero • Dario Pacciarelli •
Antonio Sforza
Editors

Optimization and Decision Science

ODS, Virtual Conference, November 19, 2020

AIRO
ASSOCIAZIONE ITALIANA DI RICERCA OPERATIVA
OPTIMIZATION AND DECISION SCIENCE

Springer

Editors
Raffaele Cerulli
Dipartimento di Matematica
Università degli Studi di Salerno
Fisciano, Italy

Mauro Dell'Amico
Dipartimento di Scienze e Metodi dell'Ingegneria
Università di Modena e Reggio Emilia
Reggio Emilia, Italy

Francesca Guerriero
DIMEG - Dipartimento di Ingegneria Meccanica, Energetica e Gestionale
Università della Calabria
Rende, Italy

Dario Pacciarelli
Dipartimento di Ingegneria
Università di Roma Tre
Roma, Italy

Antonio Sforza
Dipartimento di Informatica e Sistemistica
Università di Napoli Federico II
Napoli, Italy

ISSN 2523-7047 ISSN 2523-7055 (electronic)
AIRO Springer Series
ISBN 978-3-030-86840-6 ISBN 978-3-030-86841-3 (eBook)
https://doi.org/10.1007/978-3-030-86841-3

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

Preface

ODS2020, International Conference on "Optimization and Decision Science," organized by AIRO, the Italian Operations Research Society, was held online on November 19, 2020, due to the pandemic event, with Springer's support. In spite of the difficult situation, 60 talks were presented and almost 200 people participated in the conference.

In this volume, the reader will find the research papers submitted and accepted for publication after a peer-review process.

These 20 papers offer new and original contributions from both methodological and applied perspective, using models based on continuous and discrete optimization, graph theory, and network optimization, solved by heuristics, metaheuristics, and exact methods. A wide diversity of real-world applications is addressed. For this reason, although the book is aimed primarily at researchers and PhD students of the operations research community, the interdisciplinary content makes it interesting for scholars and researchers from other disciplines, including artificial intelligence, computer sciences, economics, mathematics, and engineering, as well as for practitioners facing complex decision-making problems in the aforementioned areas.

The 20 accepted papers are organized into 4 topical parts, listed in alphabetical order: Game Theory and Optimization; Healthcare; Scheduling and Planning; and Transportation and Logistics. In each part, the papers are listed alphabetically by the last name of the first author.

In the first part, Game Theory and Optimization, the reader will find the following chapters:

Integer Programming Reformulations in Interval Linear Programming, by Garajová et al. (2021) As known, interval linear programming provides a mathematical model for optimization problems affected by uncertainty, in which the uncertain data can be independently perturbed within the given lower and upper bounds. The authors explore the possibility of applying the existing integer programming techniques in tackling some of the problems arising in these operations.

On the Optimal Generalization Error for Weighted Least Squares Under Variable Individual Supervision Times, by Gnecco (2021) In this chapter, the trade-off between the number of labeled examples in linear regression and their precision of supervision is optimized, for the case where distinct examples can be associated with one among $M > 2$ different supervision times, and weighted least squares is used for learning.

On Braess' Paradox and Average Quality of Service in Transportation Network Cooperative Games, by Passacantando et al. (2021) In the theory of congestion games, the Braess' paradox shows that adding one resource to a network may sometimes worsen, rather than improve, the overall network performance. Here the paradox is investigated under a cooperative game theoretic setting, in contrast to the non-cooperative one typically adopted in the literature.

Optimal Improvement of Communication Network Congestion via Nonlinear Programming with Generalized Nash Equilibrium Constraints, by Passacantando and Raciti (2021) The chapter considers a popular model of congestion control in communication networks where each player/user sends their flow on a path of the network, with a cost function consisting of pricing and utility terms. The authors assume that the network system manager can invest a given amount of resource to improve the network by enhancing the capacity of a subset of links. The decision problem is modeled as a nonlinear knapsack problem with generalized Nash equilibrium constraints, giving some preliminary numerical results.

A Note on Network Games with Strategic Complements and the Katz–Bonacich Centrality Measure, by Raciti and Passacantando (2021) The paper investigates a class of network games by using the variational inequality approach. In the case where the Nash equilibrium of the game has some boundary components, they derive a formula which connects the equilibrium to the Katz-Bonacich centrality measure, thus generalizing the classical result for the interior solution case.

In the second part, Healthcare, the reader will find the following chapters:

An Optimization Model for Managing Reagents and Swab Testing During the COVID-19 Pandemic, by Colajanni et al. (2021) The authors affirm that COVID-19 pathology is characterized also by asymptomatic patients who could considerably spread the virus without being aware of it. Therefore, swab tests have to be used to diagnose positive cases. The paper presents a multi-period resource allocation model with the objective of maximizing the quantity of all analyzed swabs while minimizing the time required to obtain the swabs result, the costs due to increase the number of swabs analyzed per unit time, and the cost to transfer swabs between laboratories.

Modelling and Solving Patient Admission and Hospital Stay Problems, by Guido et al. (2021) Patient admission and patient-to-room assignment problems in a well-defined planning horizon are considered. The proposed optimization model is embedded in a matheuristic, tested by a set of benchmark instances characterized

by real-world features. The experimental results show that the solution approach is effective and allows to obtain optimal/sub-optimal solutions in short computational times.

A Two-Stage Variational Inequality for Medical Supply in Emergency Management, by Scrimali and Fargetta (2021) The chapter proposes a stochastic approach to optimizing competition between healthcare institutions for medical supplies in emergency situations, caused by natural disasters. A scenario-based stochastic programming model in a generalized Nash equilibrium framework is proposed, providing the optimal amounts of medical supplies from warehouses to hospitals, in order to minimize both the purchasing cost and the transportation costs. A two-stage stochastic programming model is proposed, taking into account the unmet demand at the first stage and the consequent penalty. An alternative two-stage variational inequality formulation is also presented.

In the third part, Scheduling and Planning, the reader will find the following chapters:

The Value of the Stochastic Solution in a Two-Stage Assembly-to-Order Problem, by Brandimarte et al. (2012) The authors consider a simple assembly to order problem, where components must be manufactured under demand uncertainty and end items are assembled only after demand is realized. The problem can be cast as a two-stage stochastic linear program with recourse. The chapter investigates the conditions under which a stochastic programming approach yields significant advantages over a straightforward deterministic model based on the expected value of demand.

Robust Optimal Planning of Waste Sorting Operations, by Pinto et al. (2021) This chapter investigates the operations of waste recycling centers where materials are collected by a fleet of trucks and then sorted in order to be converted in secondary raw materials. The chapter proposes a mixed integer linear programming model for planning and scheduling the packaging waste recycling operations taking into consideration the stochastic nature of waste arrivals. Experiments are performed on instances taken from a real case in Italy and comparisons are made against different planning strategies.

Solution Approaches for the Capacitated Scheduling Problem with Conflict Jobs, by Tresoldi (2021) This chapter presents a new arc-based mathematical formulation and a heuristic algorithm for the capacitated scheduling problem with conflicting jobs. The effectiveness of the approach is tested through extensive computational experiments.

In the fourth part, Transportation and Logistics, the reader will find:

A Decision Model for Enhancing Driving Security, by Baldi et al. (2021) Intelligent Advance Driver Assistance Systems (ADAS) can improve vehicle control performance and, thus, driver and passenger safety. In particular, identification and prediction of driving intention are fundamental for avoiding collisions as

they can provide useful information to drivers and vehicles in their vicinity. The paper proposes a lane change prediction model based on machine learning able to distinguish between left and right lane changes, a distinction that becomes particularly important when driving in a highway. Models have been trained and validated using a real dataset gathered online by using a high-tech demonstrator vehicle provided by Fiat Research Center within the European Project DESERVE. Two models based on Support Vector Machines and Random Forest are proposed.

A Two-Echelon Truck-and-Drone Distribution System: Formulation and Heuristic Approach, by Boccia et al. (2021) The authors study a two-echelon truck-and-drone distribution system where the first-echelon is composed by the depot and truck parking places, whereas the second-echelon is composed by the parking places and the final customers that are served by a fleet of drones. Starting from previous works, different truck-and-drone delivery systems have been proposed in literature, where the truck operates as a mobile depot for the drones. In this chapter, a mixed integer linear programming formulation and a two-stage heuristic that exploits the underlying structure of the problem is proposed. The approach is tested and validate on a set of instances up to 50 customers.

A Heuristic Approach for the Human Migration Problem, by Cappello et al. (2021) This chapter presents a network-based model for human migration in which a utility function is maximized. The resulting nonlinear optimization problem is characterized by a variational inequality formulation. Due to the high complexity of this problem, in order to efficiently solve realistic instances, a heuristic method is proposed. The presented algorithms are tested and compared over a number of randomly generated instances

In-store Picking Strategies for Online Orders in Grocery Retail Logistics, by Chou et al. (2021) Customers shifting from stationary to online grocery shopping and the decreasing mobility of an ageing population pose major challenges for the stationary grocery retailing sector. To fulfill the increasing demand for online grocery shopping, traditional bricks-and-mortar retailers use existing store networks to offer customers click-and-collect services. The current COVID-19 pandemic is accelerating the transition to such a mixed offline/online model, and companies are facing the need of a re-design of their business model. Currently, a majority of the operations to service online demand consists of in-store picker-to-parts order picking systems, where employees go around the shelves of the shop to pick up the articles of online orders. The paper proposes optimization ideas and solutions for these in-store operations. Experimental simulations on a real store with real online orders are performed.

An Optimization Model for the Evacuation Time in the Presence of Delay, by Daniele et al. (2021) The chapter addresses the issue of planning the emergency evacuation of occupants of a building after a disaster event like a landslide. A network model that minimizes both the travel time and the delay of evacuating is proposed, introducing also a measure of the physical difficulties of evacuees and

a parameter associated with the severity of the disaster. The variational inequality formulation is derived and a numerical example is presented.

Additive Bounds for the Double Traveling Salesman Problem with Multiple Stacks, by Diedolo and Righini (2021) The Double TSP with Multiple Stacks is a challenging combinatorial optimization problem, asking for two Hamiltonian cycles on two weighted graphs, a pick-up graph, and a delivery graph. The two cycles originate from two given depots. They visit the vertices in an order that allows a single vehicle to collect the pick-up items in a given number of stacks and to deliver them according to a Last-In-First-Out policy for each stack. The paper investigates the use of the additive bounding procedure, starting from the Held-Karp lower bound, within a branch-and-bound algorithm. Computational results show that this method can provide tighter bounds than the Double TSP relaxation.

Crowd-Shipping and Occasional Depots in the Last Mile Delivery, by Di Puglia Pugliese et al. (2021) Crowd-shipping is a new delivery paradigm that is gaining success in the last-mile and same-day delivery process. In crowd-shipping, the deliveries are carried out by both regular company vehicles and some crowd-drivers, named occasional drivers (ODs). ODs are ordinary people available to make deliveries, for a small compensation. The paper considers a setting in which a company not only has ODs available to make deliveries, but they may also use the services of intermediate pickup and delivery points, named occasional depots. In order to optimize the use of these depots, it considers two distinct groups of ODs with different operative ranges. Occasional depots are activated only if it is necessary or convenient, implying an "activation cost," which is the main difference with respect to the classical problem with transshipments nodes. These depots should increase the flexibility of the system and they lead to a more efficient managing of the uncertain availability of ODs. This chapter presents a mixed integer linear programming model able to represent this framework. Computational experiments to validate it on small size instances are carried out.

As editors of the volume, we thank the Program Committee, composed by the AIRO Scientific Board, the invited lecturer, the authors, and the researchers who spent their time for the review process, thus contributing to improve the quality of the selected papers. Finally, we express gratitude to the Springer team for support and cooperation in publishing this volume, bringing it to a nice form.

Branch and Bound and Dynamic Programming Approaches for the Path Avoiding Forbidden Pairs Problem, by Ferone et al. (2021) The chapter proposes a branch and bound and a dynamic programming algorithm for the Path Avoiding Forbidden Pairs Problem. Given a network and a set of forbidden node pairs, the problem consists in finding the shortest path from a source node s to a target node t, avoiding to traverse both nodes of any of the forbidden pairs. The problem has been shown to be NP-complete. The paper describes the problem, its mathematical model and two exact algorithms, comparing their performances against those of a commercial solver on instances with fully random graphs and grid graphs.

Revenue Management Approach for Passenger Transport Service: An Italian case study, by Guerriero et al. (2021) The main aim of the revenue management (RM) techniques is to sell the right product to the right customer, at the right time and price, to optimize the sales. RM has been successfully applied in numerous kinds of services. Recently, bus passenger transport has been deregulated and liberalized, thus companies are free to vary their prices, timetable, and routes. The use of RM could represent a key factor in a highly competitive market. The paper considers the problem of a bus transport company which operates from a given set of origins to a given set of destinations on a given time horizon. A dynamic programming formulation and a linear approximation are proposed. The linear approximation, representing the seat-allocation problem, is tested with reference to an Italian bus company. The computational experiments reveal that the proposed model could help the bus transport company to control the capacity levels, to improve customer service and bus utilization, by maximizing the revenue.

Fisciano, Italy — Raffaele Cerulli
Reggio Emilia, Italy — Mauro Dell'Amico
Rende, Italy — Francesca Guerriero
Roma, Italy — Dario Pacciarelli
Naples, Italy — Antonio Sforza

About This Book

This book collects selected contributions from the international conference "Optimization and Decision Science" (ODS2020), which was held online on November 19, 2020, and organized by AIRO, the Italian Operations Research Society.

The book offers new and original contributions on optimization, decisions science, and prescriptive analytics from both a methodological and applied perspective, using models and methods based on continuous and discrete optimization, graph theory and network optimization, analytics, multiple criteria decision-making, heuristics, metaheuristics, and exact methods.

In addition to more theoretical contributions, the book chapters describe models and methods for addressing a wide diversity of real-world applications spanning health, transportation, logistics, public sector, manufacturing, and emergency management.

Although the book is aimed primarily at researchers and PhD students in the operations research community, the interdisciplinary content makes it interesting for practitioners facing complex decision-making problems in the aforementioned areas, as well as for scholars and researchers from other disciplines, including artificial intelligence, computer sciences, economics, mathematics, and engineering.

Contents

About the Editors

Raffaele Cerulli is Full Professor of Operations Research at the University of Salerno. His main research interests focus on combinatorial optimization problems: labeled graph problems, minimum spanning tree problem, problems of the traveling salesman, vehicle routing problems, wireless sensor network, and interval linear programming. He has organized national and international conferences/schools in these fields. He is the author of almost 80 papers on discrete optimization and related areas. He is director of the Department of Mathematics at the University of Salerno and member of the scientific committee of UMI (Italian Mathematical Union). He is member of the board of the Italian Association for Operations Research (AIRO). He is editor-in-chief of *Soft Computing* (Springer) and of *Advances in Computational Intelligence* (Springer). He has participated as principal investigator in many international funded research projects.

Mauro Dell'Amico is Full Professor of Operations Research at the University of Modena and Reggio Emilia. His main research interests focus on combinatorial optimization as primarily applied to mobility, logistics, transportation, and production planning and scheduling. He is the author of /Assignment Problems/ (SIAM 2012) and more than 80 papers on discrete optimization and related areas. He is a member of the board of the Italian Association for Operations Research (AIRO) and president of the Interuniversity Consortium for Optimization and Operations Research. He has participated as principal investigator in many international funded research projects.

Francesca Guerriero graduated with honors in management engineering from the University of Calabria, Italy. She obtained her PhD in system engineering and computer science from the same University. She was visiting research fellow at the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, MA, USA. She is Full Professor of Operations Research in the Department of Mechanical, Energy and Management Engineering, University of Calabria. Francesca is currently the vice dean of the Department of Mechanical, Energy and Management Engineering, University of Calabria, and she is vice

president of the Italian Association for Operations Research (AIRO). Her main research interests are in the area of network optimization, logistics and distribution, revenue management, project management, optimization, and big data. She is co-author of more than 130 papers published in prestigious journal in the operations research field. She has been and is a member of the scientific committee of several International Conferences and of the editorial board of several scientific journals.

Dario Pacciarelli is Full Professor of Operations Research at Roma Tre University. His main research interests are in discrete optimization and scheduling theory, with application to public transport, logistics, production planning, and scheduling, among others. He is the author of more than 100 publications in journals, books, and conference proceedings. He is president of the Italian Association for Operations Research (AIRO), president of the Italian Federation of Applied Mathematics (FIMA), member of the board of the International Association of Railway Operations Research (IAROR), and member of the International Scientific Committee of CASPT—Conference on Advanced Systems for Public Transport.

Antonio Sforza formerly Full Professor of Operations Research, held courses on optimization and problem solving in the Polytechnic School of the University Federico II of Naples. His research activity is devoted to network optimization models and methods, particularly to city logistics, traffic management and control, critical infrastructure protection, and organizing in these fields national and international conferences. He is author of more than 80 publications in books, journals, and conference proceedings. He is member of the executive board of AIRO—Italian Operations Research Society—and editor of the AIRO-Springer Series.

Part I
Game Theory and Optimization

Integer Programming Reformulations in Interval Linear Programming

Elif Garajová, Miroslav Rada, and Milan Hladík

Abstract Interval linear programming provides a mathematical model for optimization problems affected by uncertainty, in which the uncertain data can be independently perturbed within the given lower and upper bounds. Many tasks in interval linear programming, such as describing the feasible set or computing the range of optimal values, can be solved by the orthant decomposition method, which reduces the interval problem to a set of linear-programming subproblems—one linear program over each orthant of the solution space. In this paper, we explore the possibility of utilizing the existing integer programming techniques in tackling some of these difficult problems by deriving a mixed-integer linear programming reformulation. Namely, we focus on the optimal value range problem, which is NP-hard for general interval linear programs. For this problem, we compare the obtained reformulation with the traditionally used orthant decomposition and also with the non-linear absolute-value formulation that serves as a basis for both of the former approaches.

Keywords Interval linear programming · Integer programming · Optimal value range

E. Garajová (✉) · M. Hladík
Charles University, Faculty of Mathematics and Physics, Department of Applied Mathematics, Prague, Czech Republic

Prague University of Economics and Business, Faculty of Informatics and Statistics, Department of Econometrics, Prague, Czech Republic
e-mail: elif@kam.mff.cuni.cz; hladik@kam.mff.cuni.cz

M. Rada
Prague University of Economics and Business, Faculty of Finance and Accounting, Department of Financial Accounting and Auditing & Faculty of Informatics and Statistics, Department of Econometrics, Prague, Czech Republic
e-mail: miroslav.rada@vse.cz

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_1

1 Introduction

Optimization under uncertainty plays a crucial role in modeling and solving real-world problems with inexact input data. In this paper, we consider the approach of interval linear programming [9, 17], which provides a suitable model for problems with uncertain data that can be independently perturbed within the given lower and upper bounds. Throughout the last years, interval programming has been used as an uncertain model for various practical optimization problems, such as transportation problems with interval data [1, 4] or portfolio optimization [2] to mention some.

Several difficult tasks in interval linear programming can be solved by decomposing the problem at hand into an exponential number of classical linear programs. This is also the idea behind the frequently used orthant decomposition method, which exploits the fact that the feasible set of an interval linear program becomes a convex polyhedron when we restrict the solution space to a single orthant [7, 16].

Here, we propose and explore an alternative approach to solving such tasks by utilizing the powerful techniques of integer programming. To illustrate the idea, we derive a (mixed) integer programming reformulation for computing the best optimal value of an interval linear program based on a non-linear absolute-value formulation of the problem [8]. A similar approach can be beneficial in solving other related problems, such as describing the set of all optimal solutions of an interval linear program [5, 12]. We conduct a computational experiment to compare the absolute-value formulation and the derived mixed-integer programming reformulation for the optimal value range problem and show their efficiency against the traditional orthant decomposition [17].

2 Interval Linear Programming

Let us first review some of the notions and notation used throughout the paper. For a comprehensive introduction to interval linear programming see [9, 17] and references therein.

Given a vector $x \in \mathbb{R}^n$, we denote by $\mathrm{diag}(x)$ the diagonal matrix with entries $\mathrm{diag}(x)_{ii} = x_i$ for $i \in \{1, \ldots, n\}$. The inequality relations on the set of matrices and vectors, as well as the absolute value operator $|\cdot|$, are understood element-wise.

Interval Data Let the symbol $\mathbb{IR}$ denote the set of all closed real intervals. Given two real matrices $\underline{A}, \overline{A} \in \mathbb{R}^{m \times n}$ satisfying $\underline{A} \leq \overline{A}$, we define an *interval matrix* $\mathbf{A} \in \mathbb{IR}^{m \times n}$ as the set

$$\mathbf{A} = [\underline{A}, \overline{A}] = \{A \in \mathbb{R}^{m \times n} : \underline{A} \leq A \leq \overline{A}\}.$$

Alternatively, an interval matrix can also be determined by the *center* A_c and *radius* A_Δ, where

$$A_c = \frac{1}{2}(\overline{A} + \underline{A}), \qquad A_\Delta = \frac{1}{2}(\overline{A} - \underline{A}). \tag{1}$$

An *interval vector* $\mathbf{a} \in \mathbb{IR}^n$ can be defined analogously as an $n \times 1$ interval matrix. In the text, we denote all interval matrices and interval vectors by bold letters.

Interval Programming For an interval matrix $\mathbf{A} \in \mathbb{IR}^{m\times n}$ and interval vectors $\mathbf{b} \in \mathbb{IR}^m$, $\mathbf{c} \in \mathbb{IR}^n$, we define an *interval linear program* (abbreviated as ILP) as the set of all linear programs in the form

$$\min c^T x \text{ subject to } Ax \leq b, \tag{2}$$

with $A \in \mathbf{A}$, $b \in \mathbf{b}$ and $c \in \mathbf{c}$. For short, we also write an interval linear program determined by the triplet $(\mathbf{A}, \mathbf{b}, \mathbf{c})$ as

$$\min \mathbf{c}^T x \text{ subject to } \mathbf{A}x \leq \mathbf{b}. \tag{3}$$

A particular linear program (2) is called a *scenario* of the interval linear program (3).

For the sake of simplicity, the formulation of an interval linear program introduced in (3) is not the most general one. Since the commonly used transformations in linear programming are not always applicable in the interval framework due to the so-called dependency problem (see e.g. [6]), different formulations of interval linear programs may have different properties. However, the approach presented in this paper can also be utilized for other types of interval linear programs in the same manner.

Feasibility and Optimality Several different concepts of feasible and optimal solutions of interval linear programs have been introduced in the literature. In this paper, we adopt the notion of weak feasibility and optimality.

A vector $x^* \in \mathbb{R}^n$ is called a *weakly feasible solution* of ILP (3), if it is a feasible solution of some scenario, i.e. if $Ax^* \leq b$ holds for some $A \in \mathbf{A}$ and $b \in \mathbf{b}$. In general, the set of all weakly feasible solutions of an ILP forms a non-convex polyhedron, which is convex in each orthant [16]. By the Gerlach theorem for interval systems of inequalities [7], a vector $x \in \mathbb{R}^n$ is a weakly feasible solution of ILP (3) if and only if it solves the non-linear system

$$A_c x \leq A_\Delta |x| + \overline{b}. \tag{4}$$

Similarly, we say that a vector $x^* \in \mathbb{R}^n$ is a *weakly optimal solution* of the ILP, if it is an optimal solution of some scenario with $A \in \mathbf{A}$, $b \in \mathbf{b}$, $c \in \mathbf{c}$. Unless stated otherwise, we use the term "feasible/optimal solution" in the context of interval programming to refer to weakly feasible and weakly optimal solutions, respectively.

Optimal Values A common approach to computing optimal values of an interval linear program is to find the best and the worst value, which is optimal for some scenario of the program.

Let $f(A, b, c)$ denote the optimal value of the linear program (2), setting $f(A, b, c) = -\infty$ for unbounded programs and $f(A, b, c) = \infty$ for infeasible programs. Then, we define *optimal value range* of interval linear program (3) as the interval $[\underline{f}, \overline{f}]$, where the best optimal value $\underline{f}$ and the worst optimal value $\overline{f}$ are

$$\underline{f}(\mathbf{A}, \mathbf{b}, \mathbf{c}) = \min\{f(A, b, c) : A \in \mathbf{A}, b \in \mathbf{b}, c \in \mathbf{c}\},$$

$$\overline{f}(\mathbf{A}, \mathbf{b}, \mathbf{c}) = \max\{f(A, b, c) : A \in \mathbf{A}, b \in \mathbf{b}, c \in \mathbf{c}\}.$$

The worst optimal value $\overline{f}$ of ILP (3) can be computed in polynomial time by solving a linear program (see [3, 15]). On the other hand, computing the best optimal value $\underline{f}$ of (3) is an NP-hard problem [17]. Since it might be difficult to compute the value exactly, methods providing a sufficiently tight approximation are also of interest [11, 13].

Orthant Decomposition As the set of all weakly feasible solutions of an interval linear program becomes a convex polyhedron when we restrict the solution space to a single orthant, we can utilize this property to solve various problems over the feasible set. This idea leads to the often used *orthant decomposition* method, which solves a given problem in interval programming by decomposing it into a set of linear programming subproblems, one for each orthant of the solution space.

Orthant decomposition can also be used to obtain the best optimal value $\underline{f}$ of ILP (3). Here, we can formulate a linear program to compute the minimum value of the objective function over the feasible set in a given orthant and then take the smallest of the computed values (see [17] for further details). An *orthant* of the solution space $\mathbb{R}^n$ can be described as the set

$$\{x \in \mathbb{R}^n : \operatorname{diag}(s)x \geq 0\}$$

for a particular sign vector $s \in \{\pm 1\}^n$. Therefore, we can compute $\underline{f}$ by solving the linear program

$$\begin{aligned} &\text{minimize } (c_c - \operatorname{diag}(s)c_\Delta)^T x \\ &\text{subject to } (A_c - A_\Delta \operatorname{diag}(s))x \leq \overline{b}, \\ &\qquad\qquad \operatorname{diag}(s)x \geq 0. \end{aligned} \tag{5}$$

for each $s \in \{\pm 1\}^n$. For a given vector s, denote by $f_s(\mathbf{A}, \mathbf{b}, \mathbf{c})$ the optimal value of program (5). Then, we obtain the best optimal value as

$$\underline{f}(\mathbf{A}, \mathbf{b}, \mathbf{c}) = \min \left\{ f_s(\mathbf{A}, \mathbf{b}, \mathbf{c}) : s \in \{\pm 1\}^n \right\}.$$

This amounts to solving (at most) 2^n linear programs to compute $\underline{f}$, with n denoting the number of variables of the ILP. Note that the number of orthants that have to be explored can be lowered, if some of the variables are known to be sign-restricted (non-negative or non-positive).

3 Integer Programming Reformulations

In this section, we build on the absolute-value characterization of the feasible set by Gerlach stated in (4). We derive a mixed-integer linear programming reformulation of the system in order to design an alternative method for computing the best optimal value of an interval linear program.

The aim is to utilize the available techniques and efficient algorithms of integer linear programming to tackle some of the difficult interval problems, such as the problem of computing the optimal value range.

Absolute-Value Formulation Instead of using the orthant decomposition, we can also restate the method for computing the best optimal value as an absolute-value program [8], which is derived from the Gerlach theorem for describing the weakly feasible set. By this result, we can compute $\underline{f}$ as the optimal value of the non-linear program

$$\begin{array}{ll} \text{minimize} & c_c^T x - c_\Delta^T |x| \\ \text{subject to} & A_c x - A_\Delta |x| \leq \overline{b}. \end{array} \tag{6}$$

We can now attempt to solve formulation (6) directly as a non-linear program, or we can further linearize the program by modeling $|x|$ via binary variables and additional linear constraints as a mixed-integer linear program.

MIP Reformulation Now, we can use the absolute-value formulation (6) to derive a mixed-integer linear program for computing the best optimal value $\underline{f}$. To do this, we apply one of the traditional ways to model absolute values in integer programs using binary variables.

Here, we split the variable x into a positive and negative part as $x = x^+ - x^-$, using the lower and upper bound on x and auxiliary binary variables y_i. Then, we model the absolute value $|x|$ by introducing a new variable $z = x^+ + x^-$, leading to

the formulation

$$\begin{aligned}
\text{minimize} \quad & c_c^T x - c_\Delta^T z \\
\text{subject to} \quad & A_c x - A_\Delta z \leq \overline{b}, \\
& x = x^+ - x^-, \\
& z = x^+ + x^-, \\
& 0 \leq x_i^+ \leq |\overline{x_i}| y_i, \qquad \forall i \in \{1, \ldots, n\}, \\
& 0 \leq x_i^- \leq |\underline{x_i}|(1 - y_i), \qquad \forall i \in \{1, \ldots, n\}, \\
& y \in \{0, 1\}^n.
\end{aligned} \tag{7}$$

Note that we can also reduce the number of variables in the model by simply substituting the expressions in terms of x^+ and x^- for the variable x and its absolute value z. Using the definition of the center and the radius of an interval matrix stated in (1), we obtain the simplified mixed-integer linear program

$$\begin{aligned}
\text{minimize} \quad & \underline{c}^T x^+ - \overline{c}^T x^- \\
\text{subject to} \quad & \underline{A} x^+ - \overline{A} x^- \leq \overline{b}, \\
& 0 \leq x_i^+ \leq |\overline{x_i}| y_i, \qquad \forall i \in \{1, \ldots, n\}, \\
& 0 \leq x_i^- \leq |\underline{x_i}|(1 - y_i), \qquad \forall i \in \{1, \ldots, n\}, \\
& y \in \{0, 1\}^n.
\end{aligned} \tag{8}$$

Further Applications Apart from computing the optimal value range, integer programming reformulations can also prove useful in solving other difficult problems in interval linear programming. A description of many important characteristics and properties of an interval linear program can be derived from the Gerlach and the Oettli–Prager theorems [7, 16], which describe the weakly feasible set via a system of absolute-value inequalities.

For example, the set of all weakly optimal solutions of ILP (3) can be described by primal feasibility, dual feasibility and strong duality as the set of x-solutions of the system

$$\begin{aligned}
& Ax \leq b, \\
& A^T y = c, \ y \leq 0, \\
& c^T x = b^T y, \\
& A \in \mathbf{A}, \ b \in \mathbf{b}, \ c \in \mathbf{c}.
\end{aligned} \tag{9}$$

Note that this is a parametric system, since there are dependencies between the two occurrences of the interval parameters that cannot be captured by a simple interval linear system (e.g. the two occurrences of the matrix $A \in \mathbf{A}$ should represent the same matrix in any considered scenario). However, we can relax these dependencies

to obtain an interval linear system (see also [5, 10] and references therein), which provides an outer approximation of the optimal solution set:

$$\mathbf{A}x \leq \mathbf{b}, \quad \mathbf{A}^T y = \mathbf{c}, \ y \leq 0, \quad \mathbf{c}^T x = \mathbf{b}^T y. \tag{10}$$

Here, we assume that the two occurrences of the interval parameters $\mathbf{A}$, $\mathbf{b}$ and $\mathbf{c}$ are independent and in a particular scenario of the system, different values from the respective interval matrices and vectors can be chosen for them. System (10) is a classical interval linear system, so we can use the description of the weakly feasible set provided by the Gerlach and the Oettli–Prager theorems, leading to the absolute-value system

$$\begin{aligned} &A_c x \leq A_\Delta |x| + \overline{b}, \\ &\overline{A}^T y \leq \overline{c}, \ \underline{A}^T y \geq \underline{c}, \ y \leq 0, \\ &|c_c^T x - b_c^T y| \leq c_\Delta^T |x| - b_\Delta^T y. \end{aligned} \tag{11}$$

For system (11), we can formulate a mixed-integer linear program in a similar way as in the problem of computing the best optimal value. The program can then be used to compute an interval enclosure of the optimal set by finding the minimal/maximal value of each x_i over (11). We can also apply various integer programming relaxations and heuristics to derive more efficient approximation techniques for the optimal set. A tight approximation of the optimal set is also essential in solving the recently proposed outcome range problem [14], which generalizes the optimal value range by introducing an additional linear outcome function to the program.

4 Computational Experiment

We conducted a computational experiment to compare the derived integer programming reformulation with the traditionally used orthant decomposition method and the non-linear absolute-value formulation for the problem of finding the best optimal value $\underline{f}$ of ILP (3). Since all of these techniques are used to compute the value $\underline{f}$ exactly, the main criterion for comparison is the elapsed computation time.

Instances We compared the different programs for computing the best optimal value on a set of (pseudo-)randomly generated feasible instances. Since the best optimal value $\underline{f}$ can always be achieved for the upper bound $\overline{b}$ of the interval right-hand-side vector $\mathbf{b}$, we only generated interval data for the constraint matrix and the objective vector. Thus, each instance is described by an interval matrix $\mathbf{A} \in \mathbb{IR}^{m \times n}$, a fixed right-hand-side vector $b \in \mathbb{R}^m$ and an interval objective vector $\mathbf{c} \in \mathbb{IR}^n$.

All of the instances are in the inequality-constrained form (3) with bounded variables satisfying $x \in [-10^6, 10^6]$. With a 0.1 probability a generated interval coefficient includes both positive and negative values (i.e., 0 belongs to the interval), otherwise the coefficient satisfies $A_\Delta \in [0, 0.2|A_c|]$ and $c_\Delta \in [0, |c_c|]$. Due to the exponential nature of the considered problem, the number of variables and the number of constraints in the generated instances was limited. We generated problem instances of 31 sizes with

$$n \in \{5, 10, 15, 20\} \text{ and } m \in \{10, 20, 50, 100, 200, 500, 1000, 2000, 5000\}.$$

For each size, 20 problem instances were generated.

Methods and Implementation Three formulations of programs for computing the exact best optimal value $\underline{f}$ were tested and compared in the experiment:

- the commonly used orthant decomposition method, solving in each orthant of the solution space (determined by a sign vector $s \in \{\pm 1\}^n$) the linear program:

$$\begin{aligned} &\text{minimize } (c_c - \operatorname{diag}(s)c_\Delta)^T x \\ &\text{subject to } (A_c - A_\Delta \operatorname{diag}(s))x \leq \overline{b}, \\ &\qquad\qquad \operatorname{diag}(s)x \geq 0, \end{aligned}$$

- the non-linear absolute-value formulation based on the Gerlach theorem:

$$\begin{aligned} &\text{minimize } c_c^T x - c_\Delta^T |x| \\ &\text{subject to } A_c x - A_\Delta |x| \leq \overline{b}, \end{aligned}$$

- and the derived mixed-integer linear programming reformulation:

$$\begin{aligned} &\text{minimize } \underline{c}^T x^+ - \overline{c}^T x^- \\ &\text{subject to } \underline{A} x^+ - \overline{A} x^- \leq \overline{b}, \\ &\qquad 0 \leq x_i^+ \leq |\overline{x_i}| y_i, \qquad \forall i \in \{1, \ldots, n\}, \\ &\qquad 0 \leq x_i^- \leq |\underline{x_i}|(1 - y_i), \qquad \forall i \in \{1, \ldots, n\}, \\ &\qquad y \in \{0, 1\}^n. \end{aligned}$$

All of the methods were implemented in Python 3.8 and Gurobi 9.1 solver was used to solve the corresponding models. The non-linear formulation (6) was modeled using the general constraints in Gurobi supporting absolute-value expressions.

Results We used the three methods to compute the best optimal value $\underline{f}$ for a total of 500 instances of inequality-constrained interval linear programs. The experiment was carried out on a computer with a 16 GB RAM and an Intel Core i7-8650U processor. The results of the experiment are summarized in Tables 1 and 2, showing the average computation time (in seconds) of each method on a set of instances of a given problem size.

Table 1 The average elapsed running time (in seconds) of the three methods on instances with the best optimal value $\underline{f}$ attained at the boundary of the bounding box. The fastest running times are indicated by bold values

n	m	# inst.	Orthant decomposition	MIP formulation	Abs. value formulation
5	10	19	**0.0027**	0.0137	0.0060
5	20	4	**0.0037**	0.0155	0.0129
10	10	20	0.0728	0.0121	**0.0059**
10	20	20	0.1161	0.0150	**0.0109**
10	50	8	0.1964	0.0380	**0.0310**
15	10	20	2.4405	0.0129	**0.0085**
15	20	20	3.9577	0.0145	**0.0135**
15	50	20	8.8081	0.0309	**0.0281**
15	100	7	13.7356	**0.1480**	0.3445
20	10	20	83.5143	0.0148	**0.0110**
20	20	20	127.2103	0.0161	**0.0155**
20	50	20	321.7968	**0.0353**	0.0392
20	100	20	602.3350	**0.1873**	0.2387
20	200	1	961.1095	**2.5959**	5.9579

Table 1 presents the results of the experiment on problems, for which the best optimal value was attained at the boundary of the bounding box. This is a subset of the instances with a lower ratio of the number of constraints to the number of variables. The resulting running times show that this class of problems can be solved very efficiently through integer programming and through the absolute-value formulation. This holds even for problems of larger size, where the orthant decomposition approach may be too time-consuming. These results also indicate that the alternative approaches may prove useful in designing methods for quickly checking (weak) unboundedness of interval linear programs.

The results in Table 2 show the average running times of the three methods on the general problems. While the orthant decomposition is faster for the smallest problems with only 5 variables, we can observe the expected behavior on larger instances, where mixed-integer programming and the absolute-value formulation show their notable advantage in efficiency over exploring all orthants of the solution space. Here, using the mixed-integer programming formulation seems to be the fastest approach, with the absolute-value general constraints being slightly behind. Although the computation becomes more time-consuming with the growing number of variables and constraints, both of these approaches still present a significant improvement over the straight-forward orthant decomposition.

5 Conclusion

We explored the applicability of integer programming methods for solving some of the difficult problems in interval linear programming. Specifically, we considered the NP-hard problem of computing the best value, which is optimal for some

Table 2 The average elapsed running time (in seconds) of the three methods on general instances. The fastest running times are indicated by bold values

n	m	# inst.	Orthant decomposition	MIP formulation	Abs. value formulation
5	10	1	**0.0027**	0.0251	0.0078
5	20	16	**0.0038**	0.0340	0.0180
5	50	20	**0.0058**	0.0427	0.0374
5	100	20	**0.0089**	0.0545	0.0617
5	200	20	**0.0155**	0.0810	0.1381
5	500	20	**0.0369**	0.1346	0.2855
5	1000	20	**0.0733**	0.3015	0.5485
5	2000	20	**0.1823**	0.6925	1.3530
5	5000	20	**0.5633**	1.9077	3.0574
10	50	12	0.1941	0.0908	**0.0905**
10	100	20	0.3439	**0.1556**	0.1585
10	200	20	0.5932	**0.2737**	0.3468
10	500	20	1.4488	**0.7055**	1.1098
10	1000	20	2.9922	**1.6090**	2.6126
10	2000	20	6.2917	**3.5878**	6.6611
10	5000	20	18.1407	**9.6666**	22.3548
15	100	13	13.8732	**2.6967**	3.4412
15	200	20	24.1656	**5.0071**	7.1228
15	500	20	61.6706	**13.1993**	19.9268
15	1000	20	135.0868	**31.7762**	52.5820
15	2000	20	300.0724	**75.4248**	119.7592
20	200	19	967.2810	**171.7611**	248.0541

scenario of a given interval linear program. Based on an absolute-value formulation of the problem, we derived a mixed-integer linear program to compute the best optimal value. The conducted computational experiments show the significant advantages of utilizing the existing integer programming solvers over the commonly used orthant decomposition method, which explores all orthants of the solution space.

Since many problems in interval optimization are difficult to solve exactly, approximation methods are also of interest. Integer programming reformulations open a new direction for deriving algorithms for tightly approximating the optimal value range of an interval linear program. Other challenging problems may also be considered, such as approximating the set of all optimal solutions or computing the outcome range for a given function over the optimal set. These difficult problems can benefit from utilizing the theoretical foundations, relaxation techniques and efficient heuristics of integer programming.

Acknowledgments E. Garajová and M. Rada were supported by the Czech Science Foundation under Grant P403-20-17529S. M. Hladík was supported by the Czech Science Foundation under Grant P403-18-04735S. E. Garajová and M. Hladík were also supported by the Charles University project GA UK No. 180420.

References

1. Cerulli, R., D'Ambrosio, C., Gentili, M.: Best and worst values of the optimal cost of the interval transportation problem. In: Optimization and Decision Science: Methodologies and Applications, Springer Proceedings in Mathematics & Statistics, pp. 367–374. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67308-0_37
2. Chaiyakan, S., Thipwiwatpotjana, P.: Mean Absolute deviation portfolio frontiers with interval-valued returns. In: Seki, H., Nguyen, C.H., Huynh, V.N., Inuiguchi, M. (eds.) Integrated Uncertainty in Knowledge Modelling and Decision Making. Lecture Notes in Computer Science, pp. 222–234. Springer, Cham (2019)
3. Chinneck, J.W., Ramadan, K.: Linear programming with interval coefficients. J. Oper. Res. Soc. **51**(2), 209–220 (2000). https://doi.org/10.1057/palgrave.jors.2600891
4. D'Ambrosio, C., Gentili, M., Cerulli, R.: The optimal value range problem for the Interval (immune) Transportation Problem. Omega **95**, 102059 (2020). https://doi.org/10.1016/j.omega.2019.04.002
5. Garajová, E., Hladík, M.: On the optimal solution set in interval linear programming. Comput. Optim. Appl. **72**(1), 269–292 (2019). https://doi.org/10.1007/s10589-018-0029-8
6. Garajová, E., Hladík, M., Rada, M.: Interval linear programming under transformations: Optimal solutions and optimal value range. Cent. Eur. J. Oper. Res. **27**(3), 601–614 (2019). https://doi.org/10.1007/s10100-018-0580-5
7. Gerlach, W.: Zur Lösung linearer Ungleichungssysteme bei Störung der rechten Seite und der Koeffizientenmatrix. Optimization **12**, 41–43 (1981). https://doi.org/10.1080/02331938108842705
8. Hladík, M.: Optimal value range in interval linear programming. Fuzzy Optim. Decis. Making **8**(3), 283–294 (2009). https://doi.org/10.1007/s10700-009-9060-7
9. Hladík, M.: Interval linear programming: a survey. In: Mann, Z.A. (ed.) Linear Programming—New Frontiers in Theory and Applications, chap. 2, pp. 85–120. Nova Science Publishers, New York (2012)
10. Hladík, M.: An interval linear programming contractor. In: Ramík, J., Stavárek, D. (eds.) Proceedings 30th International Conference on Mathematical Methods in Economics 2012, Karviná, Czech Republic, pp. 284–289 (Part I.). Silesian University in Opava, School of Business Administration in Karviná (2012)
11. Hladík, M.: On approximation of the best case optimal value in interval linear programming. Optim. Lett. **8**(7), 1985–1997 (2014). https://doi.org/10.1007/s11590-013-0715-5
12. Mishmast Nehi, H., Ashayerinasab, H.A., Allahdadi, M.: Solving methods for interval linear programming problem: a review and an improved method. Oper. Res. **20**(3), 1205–1229 (2020). https://doi.org/10.1007/s12351-018-0383-4
13. Mohammadi, M., Gentili, M.: Bounds on the worst optimal value in interval linear programming. Soft Comput. **23**(21), 11055–11061 (2019). https://doi.org/10.1007/s00500-018-3658-z
14. Mohammadi, M., Gentili, M.: The outcome range problem in interval linear programming. Comput. Oper. Res. **129**, 105160 (2021). https://doi.org/10.1016/j.cor.2020.105160
15. Mráz, F.: Calculating the exact bounds of optimal values in LP with interval coefficients. Ann. Oper. Res. **81**, 51–62 (1998). https://doi.org/1023/A:1018985914065
16. Oettli, W., Prager, W.: Compatibility of approximate solution of linear equations with given error bounds for coefficients and right-hand sides. Numer. Math. **6**(1), 405–409 (1964). https://doi.org/10.1007/BF01386090
17. Rohn, J.: Interval linear programming. In: Fiedler, M., Nedoma, J., Ramík, J., Rohn, J., Zimmermann, K. (eds.) Linear Optimization Problems with Inexact Data, pp. 79–100. Springer, Boston, (2006). https://doi.org/10.1007/0-387-32698-7_3

On the Optimal Generalization Error for Weighted Least Squares Under Variable Individual Supervision Times

Giorgio Gnecco

Abstract In this short paper, the trade-off between the number of labeled examples in linear regression and their precision of supervision is investigated and optimized, for the case in which distinct examples can be associated with one among $M > 2$ different supervision times, and weighted least squares is used for learning. The analysis extends the one made in one section of *Gnecco and Nutarelli, Optimization Letters, 2019, https://doi.org/10.1007/s11590-019-01486-x*, which was limited to the case $M = 2$. The results show that, for the specific learning problem, there is no advantage in applying weighted least squares instead of ordinary least squares as the learning algorithm.

Keywords Optimal M-tuple of supervision times · Linear regression · Variance control · Weighted least squares · Large-sample approximation of the generalization error

1 Introduction

Supervised machine learning [1, Chapter 2] and, in particular, in its context, Statistical Learning Theory (SLT) [10], provide methods and algorithms aimed to make a machine able to learn from experience. This is typically achieved by formulating and optimizing a suitable cost which quantifies the trade-off between minimizing the average prediction error on the so-called training set (i.e., a dataset made of labeled feature vectors, which are used to train the machine), and guaranteeing a sufficiently large generalization capability of the trained machine on input test examples, which were not available to it during the learning phase.

G. Gnecco (✉)
AXES Research Unit, IMT School for Advanced Studies, Lucca, Italy
e-mail: giorgio.gnecco@imtlucca.it

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_2

In this context, in some practical applications to fields such as engineering and physics, it is possible to control to some extent the noise variance of the labels. For instance, one could increase the cost (or time) associated with the supervision of each input example, possibly reducing that variance (this would happen, e.g., if more expensive measurement devices were used). In this situation, it is natural to analyze—then, to optimize—the resulting trade-off between the training sample size and its precision of supervision, with respect to a suitably-defined generalization error related to the function approximation learned by the machine. Such a trade-off was recently considered in [2], which investigated a modification of the classical linear regression problem, characterized by the following feature: given a fixed upper bound on the supervision time available for the supervision of the whole training set, one has the possibility of varying the time (hence, the cost) dedicated to the supervision of each training example, thus controlling the conditional variance of the measured label given the feature vector. However, by doing this, also the number of available supervised examples is influenced by the choice of the supervision time per example. By combining the estimates of the model parameters provided by the Ordinary Least Squares (OLS) regression algorithm with their classical large-sample approximation [9, Section 13.4.2], it was shown in [2] that the optimal choice of the supervision time per example strongly depends on the (either constant, or increasing, or decreasing) "returns to scale" of the precision of each supervision with respect to its cost.

In [3], the analysis made in [2] was refined and extended by considering also the situation in which distinct training examples can be associated with one of $M = 2$ possible supervision times per example, and one optimizes the fraction of examples assigned to one of such supervision times, constrained by an upper bound on the supervision time associated with the whole training set. For this situation, it was shown therein that OLS and an alternative algorithm—Weighted Least Squares (WLS)—which in principle is more suitable for this specific learning framework,[1] generate the same results at optimality.

In this short paper, we extend this last finding from [3] to the case $M > 2$, showing that a similar conclusion holds also in this situation. In more details, the results of the present analysis show that, for the specific learning problem, there is no advantage in applying WLS instead of OLS as the learning algorithm also in the case $M > 2$. Indeed, the set of optimal distributions of fractions of supervised examples associated with the various available supervision times contains the degenerate distribution for which only one supervision time is used, and this corresponds to homoskedastic measurement errors, for which WLS reduces to OLS. Differently from [3], the present analysis requires the application of Jensen's inequality and an

[1] For the two respective problems analyzed in [2] and in the extended framework of [3], OLS and WLS provide the best linear unbiased estimates of the vector of model parameters associated with the linear regression model, according to the well-known Gauss-Markov theorem [9, Section 9.4]. This depends on the fact that the measurement noise is homoskedastic in the framework considered in [2], whereas it is heteroskedastic in its extension considered in [3].

investigation of the consequences of various necessary optimality conditions, valid for the specific multivariate optimization problem under study.

The paper is structured as follows. Section 2 summarizes the results obtained in [2] for the optimization of the trade-off between training sample size and precision of supervision for the WLS case with $M = 2$. Section 3 extends them to the case $M > 2$. Finally, Sect. 4 presents possible extensions of the analysis, including one to another linear regression model.

2 Background

The following linear model for a static input-output relationship is considered: $y = \underline{\beta}'\underline{x}$, where $\underline{x} \in \mathbb{R}^{p\times 1}$ is a random feature vector with finite first and second moments, $\underline{\beta} \in \mathbb{R}^{p\times 1}$ is an unknown parameter vector, and $y \in \mathbb{R}$ is the dependent variable. In order to approximate the parameter vector $\underline{\beta}$ via a suitable estimate $\hat{\underline{\beta}} \in \mathbb{R}^{p\times 1}$ learned from data, it is assumed here that only a noisy training set is available, made of a finite number of supervised examples $(\underline{x}_n, \tilde{y}_n)$. Moreover, the $\underline{x}_n$ are independent and identically distributed as $\underline{x}$, and $\tilde{y}_n$ is an approximation of $y_n = \underline{\beta}'\underline{x}_n$, modeled as $\tilde{y}_n = y_n + \varepsilon_{n,\Delta T_n}$, where $\Delta T_n \in [\Delta T_{\min}, \Delta T_{\max}]$ is a positive supervision time associated with the n-th example, and $\varepsilon_{n,\Delta T_n}$ is an additive supervision noise, modeled as a random variable, independent from $\underline{x}_n$, having mean 0 and variance $\sigma_\varepsilon^2(\Delta T_n) = C(\Delta T_n)^{-\alpha}$ (the dependence on ΔT_n has been highlighted in the notation $\sigma_\varepsilon^2(\Delta T_n)$). Here, $\alpha > 0$ is a given constant, and $C > 0$ is another given (dimensional) constant (having the dimension of ΔT_n^α). For simplicity, the supervision times ΔT_n are assumed to be chosen a-priori (i.e., they do not depend on the realizations of the $\underline{x}_n$), and the number N of examples is such that $\sum_{n=1}^{N} \Delta T_n \leq T$ (being $T > 0$ an upper bound on the total supervision time), and $\Delta T_{\min} + \sum_{n=1}^{N} \Delta T_n > T$ (i.e., adding any other example makes the previous constraint $\sum_{n=1}^{N} \Delta T_n \leq T$ on the total supervision time be violated). As a consequence, different choices of the supervision times ΔT_n can correspond to different values of the number N of supervised examples.

Let $X_N \in \mathbb{R}^{N\times p}$ denote the training input data matrix, whose generic n-th row is the transpose of $\underline{x}_n$, and $\underline{\tilde{y}}_N \in \mathbb{R}^{N\times 1}$ the vector whose generic n-th element is $\tilde{y}_n$. Moreover, let $\underline{x}^{\text{test}}$ denote a new test example, distributed as $\underline{x}$ but independent from all the $\underline{x}_n$ and all the measurement errors $\varepsilon_{n,\Delta T_n}$, and let $y^{\text{test}} = \underline{\beta}'\underline{x}^{\text{test}}$ denote its uncorrupted output, which one aims to predict. Finally, let $\hat{e}^{\text{test}} = \hat{\underline{\beta}}'\underline{x}^{\text{test}} - y^{\text{test}}$ denote the prediction error on the new test example, and let $\text{Var}\left(\hat{e}^{test} \middle| X_N\right)$ be its conditional variance, conditioned on the input data matrix X_N (seeing $\underline{\tilde{y}}_N$ as a random vector, given X_N). In the following, we refer to that conditional variance as the (conditional) generalization error of the learning machine.

2.1 Optimal Generalization Error Under WLS with Two Possible Individual Supervision Times

In this subsection, we provide a summary of the analysis made in [3, end of Section 3] about the optimal generalization error for the model introduced above, when WLS is applied and there are $M = 2$ possible individual supervision times. In this framework, different supervised examples are associated with one of two a-priori given different computational times for supervision, giving rise to multiplicative heteroskedastic computational noise. More precisely, one assigns a fraction $\lambda \in [0, 1]$ of training examples[2] to the supervision time $\Delta T_{\min}$, and the remaining fraction $1 - \lambda$ of training examples to the supervision time $\Delta T_{\max}$. Then, one optimizes the choice of λ (and as a consequence, also the choice of the total number of training examples, for the given upper bound T on the total computational time used for all the supervisions). In this case, according to Gauss-Markov theorem, the best linear unbiased estimate of the parameter vector $\underline{\beta}$ is provided by WLS [9, Section 18.4], which is a natural choice as the learning algorithm. The expression of the resulting estimate is $\hat{\underline{\beta}}_{WLS} = (X'_{N(\lambda)}\Omega^{-1}X_{N(\lambda)})^{-1}X'_{N(\lambda)}\Omega^{-1}\underline{\tilde{y}}_{N(\lambda)}$, where $N(\lambda)$ is the number of training examples expressed as a function of λ, $\Omega := \mathrm{Var}\left(\underline{\varepsilon}\right)$ is the covariance matrix of the zero-mean vector $\underline{\varepsilon}$ of measurement errors (which in this case is diagonal, and has a fraction λ of its elements on the main diagonal equal to $\sigma_\varepsilon^2(\Delta T_{\min})$, and the remaining fraction $1 - \lambda$ of its elements on the main diagonal equal to $\sigma_\varepsilon^2(\Delta T_{\max})$), and $\underline{\tilde{y}}_{N(\lambda)}$ denotes the vector of measures $\tilde{y}_n$ ($n = 1, \ldots, N(\lambda)$). In this situation, neglecting discretization issues,[3] one has the approximation $N(\lambda) \simeq \frac{T}{\lambda\Delta T_{\min}+(1-\lambda)\Delta T_{\max}}$, whereas the covariance matrix of the estimate $\underline{\beta}_{WLS}$ conditioned on the training input data matrix $X_{N(\lambda)}$ is $\mathrm{Var}\left(\hat{\underline{\beta}}_{WLS}\middle|X_{N(\lambda)}\right) = (X'_{N(\lambda)}\Omega^{-1}X_{N(\lambda)})^{-1}$, and the variance of the prediction error on a new test example conditioned on $X_{N(\lambda)}$ is

$$\mathrm{Var}\left(\hat{e}_{WLS}^{test}\middle|X_{N(\lambda)}\right) = \mathrm{Tr}\left(\mathbb{E}\left\{\underline{x}\,\underline{x}'\right\}\mathrm{Var}\left(\hat{\underline{\beta}}_{WLS}\middle|X_{N(\Delta T)}\right)\right) \tag{1}$$

[2] In [3, end of Section 3], such examples are chosen randomly from the ones available in the training set. Since this choice does not actually depend on the specific realizations of the feature vectors, there is no loss of generality in assuming that these examples are chosen, instead, deterministically (e.g., that they form the fraction λ of first examples).

[3] The analysis made in [3] for a related problem in which WLS is replaced by OLS and all the supervised examples are associated with the same—although variable—supervision time shows that, in that case, the approximation error associated with discretization goes to 0 when T tends to $+\infty$. So, it is natural to neglect that error also in the present framework.

(being Tr and $\mathbb{E}$, respectively, the trace and expectation operators). Exploiting the large-sample approximation

$$\frac{X'_{N(\lambda)}\Omega^{-1}X_{N(\lambda)}}{N(\lambda)} \simeq \left(\frac{\lambda}{\sigma_\varepsilon^2(\Delta T_{\min})} + \frac{1-\lambda}{\sigma_\varepsilon^2(\Delta T_{\max})}\right)\mathbb{E}\left\{\underline{x}\,\underline{x}'\right\}, \tag{2}$$

Eq. (1) has the large-sample approximation

$$\begin{aligned}
&\mathrm{Var}\left(\hat{e}_{WLS}^{test}\middle|X_{N(\lambda)}\right)(\lambda) \qquad (3)\\
&\simeq \frac{p}{N(\lambda)}\cdot\frac{1}{\frac{\lambda}{\sigma_\varepsilon^2(\Delta T_{\min})}+\frac{1-\lambda}{\sigma_\varepsilon^2(\Delta T_{\max})}}\\
&= \frac{p}{N(\lambda)}\cdot\frac{\sigma_\varepsilon^2(\Delta T_{\min})\sigma_\varepsilon^2(\Delta T_{\max})}{\lambda\sigma_\varepsilon^2(\Delta T_{\max})+(1-\lambda)\sigma_\varepsilon^2(\Delta T_{\min})}\\
&\simeq \frac{p\sigma_\varepsilon^2(\Delta T_{\min})\sigma_\varepsilon^2(\Delta T_{\max})}{T}\cdot\frac{\lambda\Delta T_{\min}+(1-\lambda)\Delta T_{\max}}{\lambda\sigma_\varepsilon^2(\Delta T_{\max})+(1-\lambda)\sigma_\varepsilon^2(\Delta T_{\min})}\\
&= \frac{pC(\Delta T_{\min})^{-\alpha}(\Delta T_{\max})^{-\alpha}}{T}\cdot\frac{\lambda\Delta T_{\min}+(1-\lambda)\Delta T_{\max}}{\lambda(\Delta T_{\max})^{-\alpha}+(1-\lambda)(\Delta T_{\min})^{-\alpha}}, \qquad (4)
\end{aligned}$$

where the dependence on λ has been made explicit. Hence, the following optimization problem is stated to find an optimal choice of the fraction λ:

$$\begin{aligned}
&\underset{\lambda}{\text{minimize}} \qquad f(\lambda) := \frac{pC(\Delta T_{\min})^{-\alpha}(\Delta T_{\max})^{-\alpha}}{T}\cdot\frac{\lambda\Delta T_{\min}+(1-\lambda)\Delta T_{\max}}{\lambda(\Delta T_{\max})^{-\alpha}+(1-\lambda)(\Delta T_{\min})^{-\alpha}}\\
&\text{s. t. } 0\le\lambda\le 1\,. \qquad (5)
\end{aligned}$$

The optimization problem (5) can be easily solved, as its objective function is the ratio of two linear functions in λ. When solving it for various choices of α, one gets the following characterization of its optimal solutions:

(1) $\lambda = 1$ (i.e., all the training examples are associated with the same supervision time $\Delta T_{\min}$), if $0 < \alpha < 1$ ("decreasing returns of scale");
(2) $\lambda = 0$ (i.e., all the training examples are associated with the same supervision time $\Delta T_{\max}$), if $\alpha > 1$ ("increasing returns of scale");
(3) any $\lambda \in [0, 1]$, if $\alpha = 1$ ("constant returns of scale").

3 Extension to $M > 2$ Admissible Supervision Times per Example

We now consider a similar setting as in Sect. 2, in which, however, one can assign, for $m = 1, \ldots, M$, a fraction $\lambda_m \in [0, 1]$ of training examples to the fixed supervision time $\Delta T^{(m)} \in [\Delta T_{\min}, \Delta T_{\max}]$ (of course, the constraint $\sum_{m=1}^{M}\lambda_m = 1$

has to hold). Without loss of generality, it is assumed here $\Delta T^{(m_1)} < \Delta T^{(m_2)}$ for $m_1 < m_2$. Moreover, $\Delta T^{(1)} = \Delta T_{\min}$ and $\Delta T^{(M)} = \Delta T_{\max}$. Then, as before, one optimizes the choice of the fractions λ_m. In this case, the matrix Ω has, for $m = 1, \ldots, M$, a fraction λ_m of its elements on the main diagonal equal to $\sigma_\varepsilon^2(\Delta T^{(m)}) = C(\Delta T^{(m)})^{-\alpha}$, whereas the number of training examples $N(\lambda)$ is replaced by $N(\{\lambda_m\}_{m=1}^M) \simeq \frac{T}{\sum_{m=1}^M \lambda_m \Delta T^{(m)}}$. Finally, Eqs. (2.1), (1), (2) and (3) are replaced, respectively, by

$$\mathrm{Var}\left(\underline{\hat{\beta}}_{WLS} \middle| X_{N(\{\lambda_m\}_{m=1}^M)}\right) = (X'_{N(\{\lambda_m\}_{m=1}^M)} \Omega^{-1} X_{N(\{\lambda_m\}_{m=1}^M)})^{-1} ,$$

$$\mathrm{Var}\left(\hat{e}_{WLS}^{test} \middle| X_{N(\{\lambda_m\}_{m=1}^M)}\right) = \mathrm{Tr}\left(\mathbb{E}\left\{\underline{x}\,\underline{x}'\right\} \mathrm{Var}\left(\underline{\hat{\beta}}_{WLS} \middle| X_{N(\{\lambda_m\}_{m=1}^M)}\right)\right) ,$$

$$\frac{X'_{N(\{\lambda_m\}_{m=1}^M)} \Omega^{-1} X_{N(\{\lambda_m\}_{m=1}^M)}}{N(\{\lambda_m\}_{m=1}^M)} \simeq \sum_{m=1}^M \frac{\lambda_m}{\sigma_\varepsilon^2(\Delta T^{(m)})} \mathbb{E}\left\{\underline{x}\,\underline{x}'\right\} ,$$

$$\mathrm{Var}\left(\hat{e}_{WLS}^{test} \middle| X_{N(\{\lambda_m\}_{m=1}^M)}\right)(\{\lambda_m\}_{m=1}^M)) \simeq \frac{p}{N(\{\lambda_m\}_{m=1}^M)} \cdot \frac{1}{\sum_{m=1}^M \frac{\lambda_m}{\sigma_\varepsilon^2(\Delta T^{(m)})}}$$

$$\simeq \frac{p}{T} \cdot \frac{\sum_{m=1}^M \lambda_m \Delta T^{(m)}}{\sum_{m=1}^M \frac{\lambda_m}{\sigma_\varepsilon^2(\Delta T^{(m)})}} .$$

Hence, the following optimization problem is stated to find an optimal choice of the fractions λ_m:

$$\begin{aligned} \underset{\{\lambda_m\}_{m=1}^M}{\text{minimize}} \qquad & f(\{\lambda_m\}_{m=1}^M) := \frac{p}{T} \cdot \frac{\sum_{m=1}^M \lambda_m \Delta T^{(m)}}{\sum_{m=1}^M \frac{\lambda_m}{C(\Delta T^{(m)})^{-\alpha}}} \\ \text{s. t.} \quad & \sum_{m=1}^M \lambda_m = 1 , \\ & \lambda_m \geq 0, \ \text{for } m = 1, \ldots, M . \end{aligned} \tag{6}$$

We distinguish among the following three cases:

(a) $0 < \alpha < 1$. Interpreting the λ_m as probabilities of the realizations $\Delta T^{(m)}$ of a discrete random variable ΔT, the objective function of the optimization problem (6) can be written as $f(\{\lambda_m\}_{m=1}^M) = \frac{pC}{T} \cdot \frac{\mathbb{E}\{\Delta T\}}{\mathbb{E}\{(\Delta T)^\alpha\}}$, , i.e., as a positive constant times the ratio between the expected value of ΔT and the expected

value of the increasing concave function $(\Delta T)^{\alpha}$. Using Jensen's inequality for concave functions, one gets

$$\mathbb{E}\{(\Delta T)^{\alpha}\} \leq (\mathbb{E}\{\Delta T\})^{\alpha}\,, \tag{7}$$

$$f(\{\lambda_m\}_{m=1}^{M}) \geq \frac{pC}{T} \cdot \frac{\mathbb{E}\{\Delta T\}}{(\mathbb{E}\{\Delta T\})^{\alpha}} = \frac{pC}{T}(\mathbb{E}\{\Delta T\})^{1-\alpha} \geq \frac{pC}{T}(\Delta T_{\min})^{1-\alpha}\,. \tag{8}$$

Since, for $\lambda_1 = 1$ and $\lambda_m = 0$ for $m = 2, \ldots, M$, one gets

$$f(\{\lambda_m\}_{m=1}^{M}) = \frac{pC}{T}(\Delta T_{\min})^{1-\alpha}\,, \tag{9}$$

one concludes from Eqs. (8) and (9) that this is an optimal solution of the optimization problem (6). This is unique because the equality in Eq. (7) holds if and only if the random variable ΔT is almost surely constant.

(b) $\alpha > 1$. In this case, one can exploit the equivalence between the optimization problem (6) and the following one:

$$\begin{aligned} \underset{t,\{\lambda_m\}_{m=1}^{M}}{\text{minimize}} \quad & t \\ \text{s. t. } & \frac{p}{T}\sum_{m=1}^{M}\lambda_m \Delta T^{(m)} = t\sum_{m=1}^{M}\frac{\lambda_m}{C(\Delta T^{(m)})^{-\alpha}} \\ & \sum_{m=1}^{M}\lambda_m = 1\,, \\ & \lambda_m \geq 0, \ \text{ for } m = 1, \ldots, M\,. \end{aligned}$$

Introducing the Lagrangian function

$$\begin{aligned} L(t, \{\lambda_m\}_{m=1}^{M}, \mu, \nu, \{\eta_m\}_{m=1}^{M}) := t + \mu\left(\frac{p}{T}\sum_{m=1}^{M}\lambda_m \Delta T^{(m)} - t\sum_{m=1}^{M}\frac{\lambda_m}{C(\Delta T^{(m)})^{-\alpha}}\right) \\ +\nu\left(\sum_{m=1}^{M}\lambda_m - 1\right) + \sum_{m=1}^{M}\eta_m\lambda_m\,, \end{aligned}$$

one gets the following necessary conditions for optimality:[4]

$$\frac{\partial L}{\partial t} = 1 - \mu \sum_{m=1}^{M} \frac{\lambda_m}{C(\Delta T^{(m)})^{-\alpha}} = 0\,, \tag{10}$$

$$\frac{\partial L}{\partial \lambda_m} = \frac{\mu p}{T}\Delta T^{(m)} - \frac{\mu t}{C} \cdot \frac{1}{(\Delta T^{(m)})^{-\alpha}} + \nu + \eta_m = 0\,, \text{ for } m = 1, \ldots, M\,, \tag{11}$$

$$\frac{p}{T} \sum_{m=1}^{M} \lambda_m \Delta T^{(m)} = t \sum_{m=1}^{M} \frac{\lambda_m}{C(\Delta T^{(m)})^{-\alpha}}\,, \tag{12}$$

$$\sum_{m=1}^{M} \lambda_m = 1\,, \tag{13}$$

$$\lambda_m \geq 0\,, \text{ for } m = 1, \ldots, M\,, \tag{14}$$

$$\eta_m \leq 0\,, \text{ for } m = 1, \ldots, M\,, \tag{15}$$

$$\eta_m \lambda_m = 0\,, \text{ for } m = 1, \ldots, M\,. \tag{16}$$

By Eqs. (12) and (14), one has $t \geq 0$, whereas Eqs. (10) and (14) imply $\mu > 0$. Since, for each fixed pair of choices for $t \geq 0$ and $\mu > 0$, the function $g_{t,\mu}(\Delta T^{(m)}) := \frac{\mu p}{T}\Delta T^{(m)} - \frac{\mu t}{C} \cdot \frac{1}{(\Delta T^{(m)})^{-\alpha}}$ is strictly concave with respect to $\Delta T^{(m)}$, and $\eta_m = 0$ for every $\lambda_m > 0$ by Eq. (16), one concludes from Eq. (11) that there exist at most two choices $\hat{m}_1$ and $\hat{m}_2$ for m for which $\lambda_m > 0$. By Eq. (13), one gets $\lambda_{\hat{m}_2} = 1 - \lambda_{\hat{m}_1}$. Hence, the optimization problem (6) is actually reduced to the case $M = 2$ investigated in Sect. 2.1. So, the minimum objective value is actually obtained for $\hat{m}_2 = M$, $\lambda_{\hat{m}_1} = 0$, and $\lambda_{\hat{m}_2} = 1$, i.e., for the degenerate distribution with support on $\Delta T_{\max}$.

(c) $\alpha = 1$. In this case, the optimization problem (6) reduces to

$$\begin{aligned} \underset{\{\lambda_m\}_{m=1}^{M}}{\text{minimize}} \quad & \frac{p}{T} \cdot \frac{\sum_{m=1}^{M} \lambda_m \Delta T^{(m)}}{\frac{1}{C}\sum_{m=1}^{M} \lambda_m \Delta T^{(m)}} = \frac{pC}{T} \\ \text{s. t.} \quad & \sum_{m=1}^{M} \lambda_m = 1\,, \\ & \lambda_m \geq 0, \text{ for } m = 1, \ldots, M\,. \end{aligned}$$

Hence, its objective function is constant, and any choice of the λ_m compatible with its constraints is optimal.

[4] One can easily check that the Mangasarian-Fromovitz's [8] constraint qualifications hold for any admissible solution to this problem.

Table 1 Numerical results for the optimization problem (6)

Case	Optimal value of $f(\{\lambda_m\}_{m=1}^M)$	Smallest (pseudo)random value of $f(\{\lambda_m\}_{m=1}^M)$
$\alpha = 0.5$	0.0100	0.0110
$\alpha = 1.5$	0.0045	0.0046
$\alpha = 1$	0.0100	0.0100

Concluding, similarly to the case of two admissible fractions, the following optimal solutions are obtained to the optimization problem (6):

(1) $\lambda_1 = 1$, and $\lambda_m = 0$ for $m = 2, \ldots, M$ (i.e., all the training examples are associated with the same supervision time $\Delta T_{\min}$), if $0 < \alpha < 1$ ("decreasing returns of scale");
(2) $\lambda_M = 1$, and $\lambda_m = 0$ for $m = 1, \ldots, M-1$ (i.e., all the training examples are associated with the same supervision time $\Delta T_{\max}$), if $\alpha > 1$ ("increasing returns of scale");
(3) any set of fractions $\lambda_m \in [0, 1]$ for which $\sum_{m=1}^M \lambda_m = 1$, if $\alpha = 1$ ("constant returns of scale").

Since, in each of the three cases above, at least one optimal solution corresponds to a homoskedastic model for the measurement noise (indeed, all the supervised examples are associated with the same variance of the measurement noise), in that situation WLS actually reduces to OLS. So, for the learning problem investigated in the paper, there is actually no advantage in assuming from the beginning that WLS is used as a learning algorithm instead of the simpler OLS algorithm.

As a final numerical check of optimality, for each of the three cases $0 < \alpha < 1$, $\alpha > 1$, and $\alpha = 1$, the value of the objective function $f(\{\lambda_m\}_{m=1}^M)$ at any of the optimal solutions to the problem (6) is compared with its smallest value obtained on a set of $P = 100{,}000$ feasible vectors generated (pseudo)randomly as follows. Each time, using the MATLAB command `rand.m`, one generates M (almost) independent realizations $\tilde{\lambda}_r$ $(r = 1, \ldots, M)$ of a random variable, uniformly distributed on $(0, 1)$. Then, one sets $\lambda_m := \frac{\tilde{\lambda}_m}{\sum_{r=1}^M \tilde{\lambda}_r}$ $(m = 1, \ldots, M)$, to impose the constraint $\sum_{m=1}^M \lambda_m = 1$. The parameters of the problem (6) are taken as follows: $p = 10$, $M = 5$, $\Delta T^{(m)} = 1, 2, 3, 4, 5$ s, $T = 1000$ s, $C = 1\,\text{s}^\alpha$. The choices $\alpha = 0.5$ and $\alpha = 1.5$ are taken as representatives of the cases $0 < \alpha < 1$ and $\alpha > 1$, respectively. The numerical results, reported in Table 1, confirm the optimality of the solutions found by the theoretical analysis.[5]

[5] An alternative numerical check can be obtained by running a multi-start optimization algorithm starting from several (pseudo)random initializations, then comparing the smallest value of the objective function determined by the algorithm with the one assumed by the optimal solutions provided by the theoretical analysis above.

Remark 1 The optimization problem (6) can be extended to the following infinite-dimensional version, formulated through Riemann-Stieltjes integrals:

$$\underset{\Lambda}{\text{minimize}}\ f(\Lambda) := \frac{p}{T} \cdot \frac{\int_{\Delta T_{\min}}^{\Delta T_{\max}} t d\Lambda(t)}{\int_{\Delta T_{\min}}^{\Delta T_{\max}} \frac{d\Lambda(t)}{Ct^{-\alpha}}}$$

$$\text{s. t. } \Lambda \text{ is a cumulative distribution function on } [\Delta T_{\min}, \Delta T_{\max}]\,. \qquad (17)$$

By a limiting argument (allowed by the definition of Riemann-Stieltjes integral), the results of the analysis above extend directly to (17): e.g., for $0 < \alpha < 1$, the optimal probability measure is concentrated on $\Delta T_{\min}$.

4 Possible Developments

A first possible extension concerns the replacement of a large-sample approximation in the analysis with a suitable bound from statistical learning theory (valid either for WLS or for its version with truncated output), as done recently in [7][6] for the case of OLS with homoskedastic measurement error. In this way, the analysis would hold for any finite sample size. Such an extension looks feasible, being WLS equivalent to OLS applied to suitably-transformed data. As a second possible development, the analysis could be extended to the fixed effects panel data model [11]. This is a linear regression model able to represent unobserved heterogeneity in the dataset, via possibly different constants associated with distinct observational units. By making the additional assumption that different measurement noises are independent and that, for each observational unit, one can assign a fraction of supervised examples to one among M given supervision times per example, an optimization problem similar to the one considered in this paper could be formulated. Related optimization problems (in which all the supervised examples are associated with the same—though variable—supervision cost) were recently considered in [4–6], respectively for the balanced case (same number of supervised examples for each observational unit), the unbalanced case (different number), and the fixed effects generalized least squares panel data model (in which the measurements errors associated with each unit are temporally correlated).

Acknowledgments G. Gnecco is a member of the Gruppo Nazionale per l'Analisi Matematica, la Probabilità e le loro Applicazioni (GNAMPA) of the Istituto Nazionale di Alta Matematica (INdAM). The work was partially supported by the 2020 Italian project "Trade-off between Number of Examples and Precision in Variations of the Fixed-Effects Panel Data Model", funded by INdAM-GNAMPA.

[6] A preliminary version of the work [7] was presented at the International Conference on Optimization and Decision Science (ODS 2020).

References

1. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning, 2nd edn. Springer, New York (2009)
2. Gnecco, G., Nutarelli, F.: On the trade-off between number of examples and precision of supervision in regression problems. In: Proceedings of the 4th International Conference of the International Neural Network Society on Big Data and Deep Learning (INNS BDDL 2019), Sestri Levante, pp. 1–6 (2019)
3. Gnecco, G., Nutarelli, F.: On the trade-off between number of examples and precision of supervision in machine learning problems. Optim. Lett. (2019). https://doi.org/10.1007/s11590-019-01486-x
4. Gnecco, G., Nutarelli, F.: Optimal trade-off between sample size and precision of supervision for the fixed effects panel data model. In: Proceedings of the 5th International Conference on Machine Learning, Optimization & Data science (LOD 2019), vol. 11943 of Lecture Notes in Computer Science, pp. 531–542 (2020)
5. Gnecco, G., Nutarelli, F., Selvi, D.: Optimal trade-off between sample size, precision of supervision, and selection probabilities for the unbalanced fixed effects panel data model. Soft Comput. (2020). https://doi.org/10.1007/s00500-020-05317-5
6. Gnecco, G., Nutarelli, F., Selvi, D.: Optimal trade-off between sample size and precision for the fixed effects generalized least squares panel data model. Mach. Learn. (2021). https://doi.org/10.1007/s10994-021-05976-x
7. Gnecco, G., Raciti, F., Selvi, D.: A statistical learning theory approach for the analysis of the trade-off between sample size and precision in truncated ordinary least squares. In: Nikeghbali, A., Pardalos, P., Raigorodskii, A., and Rassias, M.Th. (eds.) High Dimensional Optimization and Probability: With a View Towards Data Science. Springer, forthcoming (2021)
8. Ruszczynski, A.: Nonlinear Optimization. Princeton University Press, Princeton (2006)
9. Ruud, P.A.: An Introduction to Classical Econometric Theory. Oxford University Press, Oxford (2000)
10. Vapnik, V.N.: Statistical Learning Theory. Wiley-Interscience, New York (1998)
11. Wooldridge, J.M.: Econometric Analysis of Cross Section and Panel Data. MIT Press, Cambridge (2002)

On Braess' Paradox and Average Quality of Service in Transportation Network Cooperative Games

Mauro Passacantando, Giorgio Gnecco, Yuval Hadas, and Marcello Sanguineti

Abstract In the theory of congestion games, the Braess' paradox shows that adding one resource to a network may sometimes worsen, rather than improve, the overall network performance. Here the paradox is investigated under a cooperative game-theoretic setting, in contrast to the non-cooperative one typically adopted in the literature. A family of cooperative games on networks is considered, whose utility function, defined in terms of a traffic assignment problem and the associated Wardrop equilibrium, expresses the average quality of service perceived by the network users.

Keywords Transportation networks · Transferable utility games · Braess' paradox · Traffic assignment · User equilibrium · Quality of service

1 Introduction

In the theory of congestion games [12], Braess' paradox [1] highlights why adding one resource to a network may in some cases worsen, rather than improve, the overall network performance. This phenomenon is typically explained through non-

M. Passacantando
Department of Computer Science, University of Pisa, Pisa, Italy
e-mail: mauro.passacantando@unipi.it

G. Gnecco
AXES Research Unit, IMT - School for Advanced Studies, Lucca, Italy
e-mail: giorgio.gnecco@imtlucca.it

Y. Hadas
Department of Management, Bar-Ilan University, Ramat Gan, Israel
e-mail: yuval.hadas@biu.ac.il

M. Sanguineti (✉)
Department of Bioengineering, Robotics, Informatics, and Systems Engineering (DIBRIS), University of Genova, Genova, Italy
e-mail: marcello.sanguineti@unige.it

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_3

cooperative game theory, and is related to the concept of price of anarchy [5]: the players (in this context, for example, the users of a road network), being driven by the pursuit of maximizing their own individual interests, tend to reduce social welfare (and sometimes, as an undesired consequence, they even fail to maximize their own individual interests, when compared to the case in which they behave in a more collaborative way). In case several resources are added to the network, however, such a non-cooperative approach, which does not take into account every potential interaction among the resources in all the possible contingent situations, does not allow one to quantify the average marginal contribution (be it positive or negative) of each resource to the overall network performance. This suggests the investigation of a cooperative version of Braess' paradox.

This work, which is in the same research direction as [16], studies Braess' paradox in the context of cooperative games with transferable utility on a graph [20], which can model, for example, transportation networks [6, 8]. The players can be either nodes or arcs of the graph (in this paper, they are arcs). The utility function of each such game is defined in terms of a suitable congestion measure over subgraphs associated with subsets of these nodes/arcs. Such a measure is computed by solving an instance of the classical user equilibrium problem via any traffic assignment algorithm (see, e.g., [15]). Then, the Shapley value of a node (or arc) of the network is used as a measure of its importance, in line with [8, 11]. Differently from the latter works, the goal here is to identify situations for which the Shapley value of a node/arc is negative (as a consequence of the specific choice of the utility function). In this case, the insertion of such an element to the network has a negative average marginal value. This indicates a degradation of the average network performance following its insertion, therefore the inopportunity of such an addition.

The work complements the analysis of [16] in several directions. A different choice of the utility function associated with the transportation network cooperative game is considered, which is proportional to the average quality of service perceived by its users. Moreover, a variation of the example in [16] is adopted for illustration purposes. An additional "fictitious" arc is included in the directed graph modeling the transportation network, in order to make the origin and destination nodes connected, in all its subgraphs derived from all possible coalitions of arcs. A novel numerical example is presented and additional computational issues are discussed.

The paper is structured as follows. Section 2 provides a background on cooperative games with transferable utility, transportation networks cooperative games, and Wardrop first principle, which is used to model the behavior of vehicles in the network. Section 3 details the case of a utility function based on a Wardrop equilibrium, which is proportional to the average quality of service perceived by the network users. Section 4 provides an application to a toy example. Finally, Sect. 5 is a short discussion.

2 Background

2.1 Cooperative Games with Transferable Utility

A *cooperative game with transferable utility* (*TU game*, see, e.g., [20]) is a pair (N, v), where:

- N is a set of players, and any subset $S \subseteq N$ is called a coalition;
- $v : 2^N \to \mathbb{R}$ is the utility (characteristic) function, with $v(\emptyset) = 0$; $v(S)$ represents the utility that can be achieved jointly by all the players in S, without any contribution from the players in $N \setminus S$.

In TU games, the utilities can be transferred from one player to another without any loss.

The Shapley value [18] is the most important point-solution in cooperative game theory, and corresponds to a suitable way of allocating the total utility in a "fair way" among the players. For each player $i \in N$, it is defined as

$$Sh(i) = \sum_{S \subseteq N} \frac{(|S| - 1)!(|N| - |S|)!}{|N|!} [v(S) - v(S \setminus \{i\})] .$$

It represents the average marginal contribution of each player across all possible coalitions, according to a suitable probability distribution (i.e., when players, starting from the empty coalition, enter the grand coalition randomly, in such a way that all orders are equally likely). It is worth noting that, due to the interpretation above, the Shapley value can be applied as a measure of players' importance not only in classical contexts in which the players are modeled as rational decision makers, but also in other more general situations in which this does not occur, e.g., when players are features in supervised machine learning problems [2], genes in microarray games [13], or joints in the analysis of motion capture datasets [10].

2.2 Transportation Network Cooperative Games

Consider a graph $G = (V, A)$, where

- V is the set of nodes;
- $A \subseteq V \times V$ is the set of arcs;
- W is the set of Origin-Destination pairs;
- d_w is the traffic demand of the OD pair w, and $d = (d_w)$;
- P_w is the set of paths joining the elements of the OD pair w;
- x_p is the flow on path p, and $x = (x_p)$;
- f_a is the flow on arc a, and $f = (f_a)$;

- $c_a(f)$ is the (non-negative) cost on arc a associated with the flow vector f;
- $C_p(x)$ is the (non-negative) cost on path p, equal to the sum of the costs on the arcs of the path p.

The set N of players is a given subset of arcs such that any OD pair can be served in the subgraph $(V, A \setminus N)$. This is a subset of suitable arcs chosen a priori because they are deemed to be important for the analysis of the specific network under investigation. Specifically, for any OD pair $w \in W$ there exists a path joining the elements of w in the subgraph $(V, A \setminus N)$ obtained by removing all the arcs in N from the arc set A (see Sect. 4 for an example).

2.3 Wardrop First Principle

We consider a transportation network whose arcs model one-way traffic roads and their weights the associated travel costs (e.g., travel times translated to monetary costs, combined with tolls, if present), which are functions of the respective arc flows. The number of vehicles traveling is considered so large that each vehicle contributes with an infinitesimally small amount of flow. According to *Wardrop first principle* [21], a Wardrop equilibrium state is such that no vehicle can unilaterally reduce its travel cost by shifting to another route. So, the resulting equilibrium (called Wardrop equilibrium) models the realistic case in which all the drivers behave in a selfish way. This equilibrium can be interpreted as a Nash equilibrium in the case of an infinite number of infinitesimal players (the vehicles) [9].

For any coalition $S \subseteq N$, the subgraph associated to S is

$$G(S) := (V, (A \setminus N) \cup S)\,.$$

A path flow $x(S)$ in the subgraph $G(S)$ is feasible if for any $w \in W$ the demand d_w is satisfied by using paths belonging to the set $P_w(S)$ of paths joining the OD pair w in $G(S)$.

A *Wardrop equilibrium* (or *user equilibrium*) in $G(S)$ is defined as a feasible path flow $\bar{x}(S)$ such that for any OD pair $w \in W$ and any $p \in P_w(S)$ one has

$$C_p(\bar{x}(S)) \begin{cases} = \lambda_w(S) & \text{if } \bar{x}_p(S) > 0, \\ \geq \lambda_w(S) & \text{if } \bar{x}_p(S) = 0, \end{cases}$$

where $\lambda_w(S)$ is the "equilibrium disutility" for the OD pair w, and $\bar{x}_p(S)$ is the component of $\bar{x}(S)$ which is associated with the path p. It follows from the definition of Wardrop equilibrium that, for any $w \in W$, one incurs the same cost $C_p(\bar{x}(S))$ on all the paths $p \in P_w(S)$ for which $\bar{x}_p(S) > 0$, and such a cost is smaller than or equal to the costs $C_p(\bar{x}(S))$ on all the other paths $p \in P_w(S)$ for which $\bar{x}_p(S) = 0$.

The specific value of $\lambda_w(S)$ is obtained a posteriori by imposing all the conditions above.

It is known [4, 19] that $\bar{x}(S)$ is a Wardrop equilibrium if and only if it solves the variational inequality

$$\langle C(\bar{x}(S)), x(S) - \bar{x}(S)\rangle \geq 0, \qquad \text{for any feasible path flow } x(S).$$

3 Utility Function Based on User Equilibrium

Consider the following utility function:

$$v^{ue}(S) := \sum_{w\in W} \sum_{p\in P_w(S)} \frac{\bar{x}_p(S)}{C_p(\bar{x}_p(S))} - \sum_{w\in W} \sum_{p\in P_w(\emptyset)} \frac{\bar{x}_p(\emptyset)}{C_p(\bar{x}_p(\emptyset))},$$

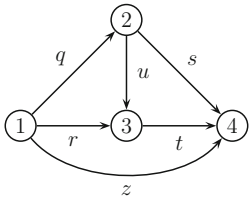
where $\bar{x}(S)$ and $\bar{x}(\emptyset)$ are Wardrop equilibria in $G(S)$ and $G(\emptyset)$, respectively. Notice that, by the definition of the Wardrop equilibrium $\bar{x}(S)$, one has $C_p(\bar{x}_p(S)) = \lambda_w(S)$ for all the paths $p \in P_w(S)$ with $\bar{x}_p(S) > 0$, so the denominators in the inner summations above do not depend on $p \in P_w(S)$. Such a utility function is well-defined when the equilibrium costs of the traveled paths are unique, which occurs when the arc costs are non-decreasing functions of the respective arc flows [3], even under possibly non-unique equilibria (indeed, for all such equilibria, one has $\sum_{p\in P_w(S)} \bar{x}_p(S) = d_w$).

The utility function introduced above is inspired by a measure of network performance versus efficiency for congested networks, which was considered in [14], but not in a cooperative setting therein. Equivalently, the term $\bar{x}_p(S)/C_p(\bar{x}_p(S))$ in the utility function $v^{ue}(S)$ represents the product between the flow $\bar{x}_p(S)$ which is served by path $p \in P_w(S)$ in the Wardrop equilibrium $\bar{x}(S)$, and the "quality of service" $1/C_p(\bar{x}_p(S))$ perceived by its vehicles. In the present work the served demand does not depend on S (indeed, even the empty coalition is able to serve it—possibly inefficiently—by using the arcs belonging to $A \setminus N$). Hence, the utility function $v^{ue}(S)$ is proportional to the improvement in the average quality of service one gets when one moves from the empty coalition to the coalition S, i.e., when the arcs in S are included in the transportation network.

Finally, it is worth remarking that the chosen utility function is not generally monotone (i.e., it is not necessarily true that $v(S) \leq v(T)$ for any $S \subseteq T \subseteq N$). Hence, the Shapley value of some arcs may be negative and the Braess' paradox can occur (see Sect. 4). This follows from the interpretation of the Shapley value as average marginal contribution of a player to the utility of a randomly generated coalition.

4 An Illustrative Example

Consider the following network with one OD pair $w = (1, 4)$ with demand d.



There are four paths connecting the OD pair:
$p_1 = (q, s)$, $p_2 = (r, t)$, $p_3 = (q, u, t)$, $p_4 = z$

We assume that the arc cost functions are defined as follows:

$$c_q = 9f_q, \quad c_r = f_r + 50, \quad c_s = f_s + 50, \quad c_t = 9f_t, \quad c_u = f_u + 10, \quad c_z = 40d + 50.$$

Hence, the path cost functions are:

$$C_1(x) = 10x_1 + 9x_3 + 50, \qquad C_2(x) = 10x_2 + 9x_3 + 50,$$
$$C_3(x) = 9x_1 + 9x_2 + 19x_3 + 10, \qquad C_4(x) = 40d + 50.$$

The arc z represents an additional "fictitious" arc, not present in the topology of the original Braess' network, which has been included here in order to make all the demand served, independently of the specific coalition S. Hence, we consider a TU game where the set of players is $N := A \setminus \{z\}$. In such a way, being the demand always served, negative Shapley values will arise only as a consequence of a deterioration of the average quality of service perceived by the network users.

The user equilibrium $\bar{x}(S)$ and the disutility $\lambda(S)$ for each coalition $S \subseteq N$ have the expressions reported in Table 1.

From the computational point of view, it can be observed that, if the flow f_a on an arc $a \in S$ is equal to 0 in correspondence of the Wardrop equilibrium $\bar{x}(S)$ for the subgraph $G(S)$, then $\bar{x}(S)$ is a Wardrop equilibrium also for the subgraph $G(S \setminus \{a\})$ obtained by removing the arc a from it. In Table 1 this occurs, e.g., for $S = \{q, r, t\}$: indeed the arc q has 0 flow in $G(S)$, because in that subgraph there is no path that connects the OD pair and uses arc q. It occurs also for $S = \{q, r, t, u\}$ when $d \leq 4$: the arc r is not used in such a case, because only the flow x_3 on path $p_3 = (q, u, t)$ is different from 0 in $\bar{x}(S)$. These arguments could help speeding up the evaluation of the utility function (and, as a consequence, of the Shapley value) for larger networks, and could be combined with empirical approximations of the Shapley value based on a subset of sampled coalitions (as done in [7], for a different and easier to compute utility function), or even based on approximate solutions of

Table 1 User equilibrium $\bar{x}(S)$ and disutility $\lambda(S)$ for each coalition $S \subseteq N$

Coalition S	$\bar{x}(S)$	$\lambda(S)$
$\emptyset$	$(0, 0, 0, d)$	$40d + 50$
$\{q\}, \{r\}, \{s\}$ $\{t\}, \{u\}$ $\{q, r\}, \{q, t\}$ $\{q, u\}, \{r, s\}$ $\{r, u\}, \{s, t\}$ $\{s, u\}, \{t, u\}$ $\{q, r, u\}$ $\{r, s, u\}$ $\{s, t, u\}$	$(0, 0, 0, d)$	$40d + 50$
$\{q, s\}$ $\{q, r, s\}$ $\{q, s, u\}$ $\{q, s, t\}$ $\{q, r, s, u\}$	$(d, 0, 0, 0)$	$10d + 50$
$\{r, t\}$ $\{q, r, t\}$ $\{r, s, t\}$ $\{r, t, u\}$ $\{r, s, t, u\}$	$(0, d, 0, 0)$	$10d + 50$
$\{q, t, u\}$	$(0, 0, d, 0)$	$19d + 10$
$\{q, r, s, t\}$	$(d/2, d/2, 0, 0)$	$5d + 50$
$\{q, s, t, u\}$	$\begin{cases} (0, 0, d, 0) & \text{if } d \leq 4 \\ (\frac{10d-40}{11}, 0, \frac{d+40}{11}, 0) & \text{if } d \geq 4 \end{cases}$	$\begin{cases} 19d + 10 & \text{if } d \leq 4 \\ \frac{109d+510}{11} & \text{if } d \geq 4 \end{cases}$
$\{q, r, t, u\}$	$\begin{cases} (0, 0, d, 0) & \text{if } d \leq 4 \\ (0, \frac{10d-40}{11}, \frac{d+40}{11}, 0) & \text{if } d \geq 4 \end{cases}$	$\begin{cases} 19d + 10 & \text{if } d \leq 4 \\ \frac{109d+510}{11} & \text{if } d \geq 4 \end{cases}$
$\{q, r, s, t, u\}$	$\begin{cases} (0, 0, d, 0) & \text{if } d \leq 4 \\ \left(\frac{5d-20}{6}, \frac{5d-20}{6}, \frac{40-4d}{6}, 0\right) & \text{if } d \in [4, 10] \\ (\frac{d}{2}, \frac{d}{2}, 0, 0) & \text{if } d \geq 10 \end{cases}$	$\begin{cases} 19d + 10 & \text{if } d \leq 4 \\ \frac{7d+230}{3} & \text{if } d \in [4, 10] \\ 5d + 50 & \text{if } d \geq 10 \end{cases}$

the variational inequalities that define the Wardrop equilibrium $\bar{x}(S)$ for different coalitions S.

Since there is a unique OD pair, the utility function v^{ue} has the following form:

$$v^{ue}(S) = \frac{d}{\lambda(S)} - \frac{d}{\lambda(\emptyset)}.$$

Moreover, it follows from Table 1 that the Shapley value of arc u is given by the following explicit formula:

$$Sh(u) = \begin{cases} \frac{d}{3(19d+10)} - \frac{d}{30(40d+50)} - \frac{d}{10(10d+50)} - \frac{d}{5(5d+50)} & \text{if } d \le 4, \\ \frac{d}{30(19d+10)} + \frac{11d}{10(109d+510)} + \frac{3d}{5(7d+230)} & \\ \quad - \frac{d}{30(40d+50)} - \frac{d}{10(10d+50)} - \frac{d}{5(5d+50)} & \text{if } d \in [4, 10], \\ \frac{d}{30(19d+10)} + \frac{11d}{10(109d+510)} - \frac{d}{30(40d+50)} - \frac{d}{10(10d+50)} & \text{if } d \ge 10. \end{cases}$$

The Shapley value of arc u is negative (i.e., a sort of cooperative version of Braess' paradox occurs) for $d \in (3.57, 7.67)$. This conclusion is similar to the one obtained in [16], where a different utility function—still based on Wardrop equilibria—was considered in the analysis. The Shapley values of the other arcs are positive for any $d > 0$ and, because of the symmetry of the arc cost functions, arcs q and t have the same Shapley value for any demand, and the same fact holds for arcs r and s. Figure 1 shows the Shapley value of each arc as a function of the traffic demand.

However, it can be verified (using similar expressions for the user equilibria and disutilities for each coalition as the ones reported in Table 1) that no negative Shapley value occurs if the arc cost function of arc u is modified to $c_u = 10 f_u + 50$ (as a consequence, e.g., of the introduction of a suitable congestion pricing scheme).

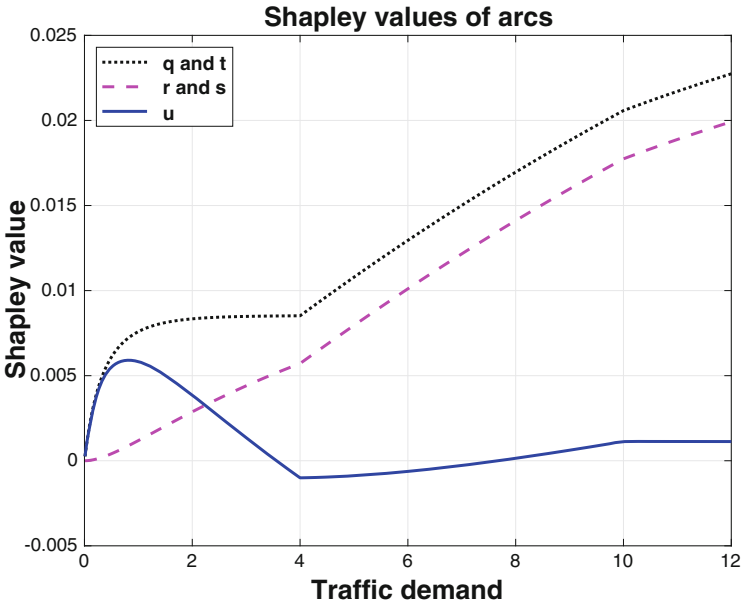


Fig. 1 Shapley value of each arc as a function of the traffic demand

It is also worth noting that, likewise in [16], no negative Shapley value is obtained if, in the definition of the utility associated with each coalition S, the Wardrop equilibrium $\bar{x}(S)$ is replaced by a flow vector $\hat{x}(S)$ maximizing the expression $\sum_{w\in W}\sum_{p\in P_w(S)}\frac{x(S)}{C_p(x(S))}$ over all flow vectors $x(S)$ that are feasible in $G(S)$, and $\bar{x}(\emptyset)$ is replaced by a vector $\hat{x}(\emptyset)$ maximizing the expression $\sum_{w\in W}\sum_{p\in P_w(\emptyset)}\frac{x(\emptyset)}{C_p(x(\emptyset))}$ over all flow vectors $x(\emptyset)$ that are feasible in $G(\emptyset)$. In this case, indeed, the resulting *system optimum* utility function

$$v^{so}(S) := \sum_{w\in W}\sum_{p\in P_w(S)}\frac{\hat{x}_p(S)}{C_p(\hat{x}_p(S))} - \sum_{w\in W}\sum_{p\in P_w(\emptyset)}\frac{\hat{x}_p(\emptyset)}{C_p(\hat{x}_p(\emptyset))}$$

is monotone, so no negative Shapley value can occur, being the Shapley value the average marginal utility of a player when it is inserted in a suitably randomly generated coalition.

5 Discussion

A first future research direction of the work is aimed at further investigating the use of congestion pricing as a way to deal with negative Shapley values. For instance, in the case of the occurrence of negative Shapley values, one could be interested in finding the minimal amount of change in the arc cost functions (induced by congestion pricing) able to make all Shapley values non-negative. A second possible extension consists in reducing the computational effort in the evaluation of the Shapley values, making it possible to analyze realistic networks characterized by a large number of arcs and various traffic demands. Indeed, in such cases, closed forms expressions of the Shapley values could be not available, or their exact evaluation could be computationally expensive. However, even a sufficiently accurate approximate evaluation of the Shapley values would be enough to achieve the final goal of detecting arcs with negative such values. A promising approach in this direction appears to be the application of supervised machine learning techniques [17] which, based on a suitable set of supervised training pairs—e.g., depending on the context, input/output pairs of the form (input vector of arc cost functions $c_a(f)$, output vector of Shapley values $Sh(i)$) or (input vector of traffic demands d, output vector of Shapley values $Sh(i)$)—could allow one to predict the output vectors of Shapley values associated with test examples (not used in the training phase), starting from the corresponding input vectors. Moreover, the possibility of guaranteeing a good generalization capability of the resulting trained machines could be investigated via a sensitivity analysis of Wardrop equilibria with respect to a change in the vector of arc cost functions.

Acknowledgments G. Gnecco, M. Passacantando and M. Sanguineti are members of the Gruppo Nazionale per l'Analisi Matematica, la Probabilità e le loro Applicazioni (GNAMPA–National Group for Mathematical Analysis, Probability and their Applications) of the Istituto Nazionale di Alta Matematica (INdAM–National Institute of Higher Mathematics). G. Gnecco and M. Sanguineti acknowledge support from the Università Italo Francese (project GALILEO 2021 no. G21_89). M. Sanguineti was partially supported by the Project PDGP 2018/20 DIT.AD016.001 "Technologies for Smart Communities" of INM (Institute for Marine Engineering) of CNR (National Research Council of Italy), where he is Research Associate. He is also Affiliated Resercher at IIT—Italian Institute of Technology (Advanced Robotics Research Line), Genova, and Visiting Professor at IMT—School for Advances Studies (AXES Research Unit), Lucca.

References

1. Braess, D.: Über ein Paradoxon aus der Verkehrsplanung. Unternehmensforschung **12**, 258–268 (1968)
2. Cohen, S., Dror, G., Ruppin, E.: Feature selection via coalitional game theory. Neural Comput. **19**, 1939–1961 (2007)
3. Correa, J.R., Stier-Moses, N.E.: Wardrop equilibria. In: J.J. Cochran (ed.) Encyclopedia of Operations Research and Management Science. Wiley, Hoboken (2011)
4. Dafermos, S.: Traffic equilibrium and variational inequalities. Transp. Sci. **14**, 42–54 (1980)
5. Dubey, P.: Inefficiency of Nash equilibria. Math. Oper. Res. **11**, 1–8 (1986)
6. Gnecco, G., Hadas, Y., Sanguineti, M.: Some properties of transportation network cooperative games. Networks **74**, 161–173 (2019)
7. Gnecco, G., Hadas, Y., Sanguineti, M.: Public transport transfers assessment via transferable utility games and Shapley value approximation. Transp. A Transp. Sci. **17**, 540–565 (2021)
8. Hadas, Y., Gnecco, G., Sanguineti, M.: An approach to transportation network analysis via transferable utility games. Transp. Res. B Methodol. **105**, 120–143 (2017)
9. Haurie, A., Marcotte, P.: On the relationship between Nash-Cournot and Wardrop equilibria. Networks **15**, 295–308 (1985)
10. Kolykhalova, K., Gnecco, G., Sanguineti, M., Volpe, G., Camurri, A.: Automated analysis of the origin of movement: an approach based on cooperative games on graphs. IEEE Trans. Hum. Mach. Syst. **50**, 550–560 (2020)
11. Michalak, T., Aadithya, K., Szczepanski, P., Ravidran, B., Jennings, N.: Efficient computation of the Shapley value for game-theoretic network centrality. J. Artif. Intell. Res. **46**, 607–650 (2013)
12. Monderer, D., Shapley, L.S.: Potential games. Games Econ. Behav. **14**, 124–143 (1996)
13. Moretti, S., Patrone, F., Benassi, S.: The class of microarray games and the relevance index for genes. TOP **12**, 256–280 (2007)
14. Nagurney, A., Qiang, Q.: A network efficiency measure with application to critical infrastructure networks. J. Glob. Optim. **40**, 261–275 (2008)
15. Panicucci, B., Pappalardo, M., Passacantando, M.: A path-based double projection method for solving the asymmetric traffic network equilibrium problem. Optim. Lett. **1**, 171–185 (2007)
16. Passacantando, M., Gnecco, G., Hadas, Y., Sanguineti, M.: Braess' paradox: a cooperative game-theoretic point of view. Networks. **78**, 264–283 (2021)
17. Shalev-Shwartz, S., Ben-David, S.: Understanding Machine Learning: From Theory to Algorithms. University Press, Cambridge (2014)
18. Shapley, L.S.: A value for n-person games. In: Kuhn, H.W., Tucker, A.W. (eds.) Contributions to the Theory of Games, pp. 307–318. Princeton University Press, Princeton (1953)

19. Smith, M.J.: The existence, uniqueness and stability of traffic equilibria. Transp. Res. B Methodol. **13**, 295–304 (1979)
20. Tijs, S.: Introduction to Game Theory. Hindustan Book Agency, New Delhi (2003)
21. Wardrop, J.G.: Some theoretical aspects of road traffic research. Proc. Inst. Civil Eng. **1**, 325–362 (1952)

Optimal Improvement of Communication Network Congestion via Nonlinear Programming with Generalized Nash Equilibrium Constraints

Mauro Passacantando and Fabio Raciti

Abstract We consider a popular model of congestion control in communication networks within the theory of generalized Nash equilibrium problems with shared constraints, where each player is a user who has to send his/her flow over a path in the network. The cost function of each player consists of two parts: a pricing and a utility term. Within this framework we assume that the network system manager can invest a given amount of money to improve the network by enhancing the capacity of its links and, because of limited financial resources, has to make a choice as to which of the links to improve. This choice is made with the help of a performance function which is computed for each set of improvements under consideration and has the property that, once the equilibrium has been reached, maximizes the aggregate utility and minimizes the sum of delays at the links. We model this problem as a nonlinear knapsack problem with generalized Nash equilibrium constraints and show some preliminary numerical experiments.

Keywords Generalized Nash equilibrium · Congestion control · Investment optimization

1 Introduction

Routing and congestion control problems have been two crucial aspects in the use of the Internet from its beginning and have gained even more importance in the recent years due to the huge increase of flows to be processed in this *big data* era. In this respect, the use of game theory has proved to be a useful tool and a large number

M. Passacantando (✉)
Department of Computer Science, University of Pisa, Pisa, Italy
e-mail: mauro.passacantando@unipi.it

F. Raciti
Department of Mathematics and Computer Science, University of Catania, Catania, Italy
e-mail: fraciti@dmi.unict.it

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_4

of papers have been devoted to model the above mentioned problems within the *cadre* of Nash equilibrium problems (see, e.g., [1, 2, 11, 13]). In this note, we focus on the congestion control framework put forward in [1], where each network user is considered as a player endowed with a cost function which is the difference of a pricing and a utility term. The pricing term has the role of congestion control, while the utility term expresses the user satisfaction. The bandwidth is the main resource of the system and players compete to send their flow from a given origin to a certain destination node. Because users share some network links, the strategy space of each player also depends on the variables of all the other players. Nash games of this kind have been introduced a long time ago by Rosen in his influential paper [12] and have been reformulated more recently by using the powerful tools of variational inequalities (see, e.g., [3, 4, 8, 10]). In the recent literature, they are termed as generalized Nash equilibrium problems (GNEPs) with shared constraints.

In this note, we adopt the model in [1] with some modifications in the pricing part of the players' cost functions, which give rise to multiple solutions of the game, but to only one *variational equilibrium*, which is considered a particularly recommended kind of equilibrium from the socio-economic standpoint (see, e.g., [3]). We then consider the possibility that the system manager can make an investment in order to improve the network performance by enhancing the capacity of the links. However, because of budget constraints not all the capacities can be enhanced and he/she has to make a decision as to which links is better to improve. The decision process is made according to its impact on a network cost function associated to each set of improvements, which has the role of maximizing the aggregate utility while minimizing the total delay at the links. The computation of the network cost function requires the knowledge of the variational Nash equilibrium in each case. Once the set of variational equilibria is known, for all the scenarios under consideration, we are then faced with a knapsack-type problem which, for instances of reasonable dimensions, can be solved by classifying all the solutions according to their corresponding relative variation of the above mentioned function.

The paper is structured as follows. In the following Sect. 2 we summarize the congestion control model proposed in [1], along with our modifications, and briefly recall some results about generalized Nash equilibrium problems with shared constraints and the variational inequality approach. In Sect. 3 we present our investment optimization model, while Sect. 4 is devoted to some illustrative numerical experiments. We conclude the paper with a small section where we touch on some possible extensions.

2 The Congestion Control Model and Its Variational Inequality Formulation

Throughout the paper, vectors of $\mathbb{R}^n$ are thought of as rows but in matrix operations they will be considered as columns and the superscript ^{T} will denote transposition. We now describe the network topology which consists of a set of links $\mathcal{L} =$

$\{l_1, \ldots, l_L\}$ connecting the nodes in the set $\mathcal{N} = \{n_1, \ldots, n_N\}$. The set of network users (players) is denoted by $\{1, \ldots, M\}$. A route R in the network is a set of consecutive links and each user i wishes to send a flow x_i between a given pair $O_i - D_i$ of origin-destination nodes; $x \in \mathbb{R}^M$ is the (route) flow of the network and the useful notation $x = (x_i, x_{-i})$ will be used in the sequel when it is important to distinguish the flow component of player i from all the others. We assume that the routing problem has already been solved and that there is only one route R_i assigned to user i. Each link l has a fixed capacity $C_l > 0$, so that user i cannot send a flow greater than the minimum capacity of the links of his/her route, and we group these capacities into a vector $C \in \mathbb{R}^L$. To describe the link structure of each route, it is useful to introduce the link-route matrix whose entries are given by:

$$A_{li} = \begin{cases} 1, & \text{if link } l \text{ belongs to route } R_i, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Using the link-route matrix, the set of feasible flows can be written in compact form as

$$X := \left\{ x \in \mathbb{R}^M : \; x \geq 0, \;\; Ax \leq C \right\}.$$

Because users share some links, the possible amount of flow x_i depends on the flows sent by the other users and is bounded from above by the quantity

$$m_i(x_{-i}) = \min_{l \in R_i} \left\{ C_l - \sum_{j=1,\; j \neq i}^{M} A_{lj} x_j \right\} \geq 0.$$

In this model the cost function of each player i has the following structure:

$$J_i(x) = P_i(x) - U_i(x_i), \tag{2}$$

where U_i represents the utility function of player i which only depends on the flow that he/she sends through the network, while P_i is a pricing term which represents some kind of toll that user i pays to exploit the network resources and depends on the flows of the players with common links to i. Players compete in a non-cooperative manner, as it is assumed that they do not communicate, and act selfishly to increase their flow. With these assumptions, the solution concept adopted is the Nash equilibrium *à la Rosen* [12], which in the modern literature is known as generalized Nash equilibrium (with shared constraints). More precisely, we say that $x^* \in \mathbb{R}^M$ is a generalized Nash equilibrium if for each $i \in \{1, \ldots, M\}$:

$$J_i(x_i^*, x_{-i}^*) = \min_{0 \leq x_i \leq m_i(x_{-i}^*)} J_i(x_i, x_{-i}^*). \tag{3}$$

It is well known (see, e.g., [5]) that, under standard differentiability and convexity assumptions, the above problem is equivalent to a quasivariational inequality and that a particular subset of solutions (called variational equilibria) can be found by solving the variational inequality $VI(F, X)$, where X is the feasible set previously defined and F is the so-called *pseudogradient* of the game, defined by:

$$F(x) = \left(\nabla_{x_1} J_1(x), \ldots, \nabla_{x_M} J_M(x)\right). \tag{4}$$

In this note we do not posit assumptions on general functions U_i and P_i, but instead consider the specific functional form treated in [1], with a slight modification, and show the existence of a unique variational equilibrium of the game. Due to our modification, it is possible that some of the capacities are saturated at equilibrium and the case that some users have zero flow at equilibrium is not ruled out. Moreover, we provide examples where also non-variational equilibria are possible. In these regards our results are in contrast with the ones in [1].

Specifically, the utility function U_i of player i is given by:

$$U_i(x_i) = u_i \log(x_i + 1), \tag{5}$$

where u_i is a parameter, while the route price function P_i of player i is the sum of the price functions of the links associated to route R_i:

$$P_i(x) = \sum_{l \in R_i} P_l \left(\sum_{j=1}^{M} A_{lj} x_j \right). \tag{6}$$

Let us notice that P_l is modeled so as to only depend on the variables of players who share the link l, namely:

$$P_l \left(\sum_{j=1}^{M} A_{lj} x_j \right) = \frac{k}{C_l - \sum_{j=1}^{M} A_{lj} x_j + e}, \tag{7}$$

where $k > 0$ is a network parameter, and e is a small positive number which we introduce to allow capacity saturation, while obtaining a well behaved function. The price function of player i is thus given by:

$$P_i(x) = \sum_{l \in R_i} \frac{k}{C_l - \sum_{j=1}^{M} A_{lj} x_j + e}, \tag{8}$$

and the resulting expression of the cost for player i is:

$$J_i(x) = \sum_{l \in R_i} \frac{k}{C_l - \sum_{j=1}^{M} A_{lj} x_j + e} - u_i \log(x_i + 1). \tag{9}$$

The following properties of the above functions are easy to check:

(1) U_i is twice continuously differentiable, non-decreasing and strongly concave on any compact interval $[0, b]$ (the last condition means that there exists $\tau > 0$ such that $\partial^2 U_i(x_i)/\partial x_i^2 \leq -\tau$ for any $x_i \in [0, b]$);
(2) P_i is twice continuously differentiable, convex and $P_i(\cdot, x_{-i})$ is non-decreasing. These properties of U_i and P_i entail an important monotonicity property of the pseudogradient F defined in (4), as the following theorem shows.

Theorem 1 *Let U_i and P_i be given as in (5) and (8), then F is strongly monotone on X, i.e., there exists $\alpha > 0$ such that*

$$(F(x) - F(y))^\top (x - y) \geq \alpha \|x - y\|^2, \qquad \forall\, x, y \in X.$$

Proof Similarly to [1], it can be shown that the Jacobian matrix of F is positive definite on X, uniformly with respect to x, thus F is strongly monotone on X. □

The unique solvability of $VI(F, X)$ is based on standard arguments, as the following theorem shows.

Theorem 2 *There exists a unique variational equilibrium of the GNEP.*

Proof The variational equilibria of the GNEP are the solutions of $VI(F, X)$. Existence of solutions of $VI(F, X)$ follows from the continuity of F and the compactness and convexity of X. The solution is unique because F is strongly monotone on X. □

We now introduce a function f which describes a global property of the game:

$$f(x) = \sum_{l \in \mathcal{L}} P_l \left(\sum_{j=1}^{M} A_{lj} x_j \right) - \sum_{i=1}^{M} U_i(x_i), \tag{10}$$

which represents the aggregate delay at the links minus the sum of the utilities of all players. The function f turns out to be the potential of the GNEP, as the following theorem shows.

Theorem 3 *The unique variational equilibrium of the GNEP coincides with the optimal solution of the system problem:* $\min_{x \in X} f(x)$.

Proof Since both f and X are convex, $\bar{x}$ is a minimizer of f on X if and only if

$$\nabla f(\bar{x})^\top (y - \bar{x}) \geq 0, \quad \forall y \in X.$$

Since $\nabla f = F$, the expression above is nothing else that the variational inequality $VI(F, X)$ which gives the variational equilibrium. □

3 The Optimal Network Improvement Model

We now suppose that the network system manager has a budget B available to improve the network performance. He/she can only increase the capacity of a subset $\widetilde{\mathcal{L}} \subseteq \mathcal{L}$ of links and knows that I_l is the investment required to enhance the capacity of link l by a given ratio γ_l. Since the available budget is generally not sufficient to enhance the capacities of all the links of $\widetilde{\mathcal{L}}$, he/she has to decide which subset of links to invest in, in order to improve as much as possible the system cost f computed at the variational equilibrium of the game with new link capacities, while satisfying the budget constraint. This problem can be formulated as an integer nonlinear program.

To this end, we define a binary variable y_l, for any $l \in \widetilde{\mathcal{L}}$, which takes on the value 1 if the investment is actually carried out on link l, and 0 otherwise. A vector $y = (y_l)_{l \in \widetilde{\mathcal{L}}}$ is feasible if the budget constraint $\sum_{l \in \widetilde{\mathcal{L}}} I_l y_l \leq B$ is satisfied. Given a feasible vector y, the new capacity of each link $l \in \widetilde{\mathcal{L}}$ is equal to

$$C_l'(y) := \gamma_l C_l y_l + (1 - y_l) C_l,$$

i.e., $C_l'(y) = \gamma_l C_l$ if $y_l = 1$ and $C_l'(y) = C_l$ if $y_l = 0$. The network manager aims to maximize the percentage relative variation of the system cost defined as

$$\varphi(y) = 100 \cdot \frac{f(\bar{x}(0)) - f(\bar{x}(y))}{|f(\bar{x}(0))|},$$

where f is defined in (10), $\bar{x}(0)$ is the variational equilibrium of the GNEP before the investment, while $\bar{x}(y)$ is the variational equilibrium of the GNEP on the improved network according to y. Therefore, the proposed optimization model is

$$\begin{aligned} \max \quad & \varphi(y) \\ \text{subject to} \quad & \sum_{l \in \widetilde{\mathcal{L}}} I_l y_l \leq B, \\ & y_l \in \{0, 1\} \quad l \in \widetilde{\mathcal{L}}. \end{aligned} \tag{11}$$

The above model can be considered a generalized knapsack problem since the computation of the nonlinear function φ at a given y requires to find the variational equilibria of the GNEPs both for the original and the improved network.

We remark that, since the variational equilibria of the GNEPs are the minimizers of f (see Theorem 3), problem (11) can be reformulated as the following mixed

integer nonlinear program:

$$\begin{aligned}
\min \quad & \sum_{l\in\widetilde{\mathcal{L}}} \frac{k}{\gamma_l C_l y_l + (1-y_l)C_l - \sum_{i=1}^{M} A_{li}x_i + e} + \sum_{l\in\mathcal{L}\setminus\widetilde{\mathcal{L}}} \frac{k}{C_l - \sum_{i=1}^{M} A_{li}x_i + e} \\
& - \sum_{i=1}^{M} u_i \log(x_i+1) \\
\text{subject to} \quad & \sum_{i=1}^{M} A_{li}x_i \le \gamma_l C_l y_l + (1-y_l)C_l \qquad \forall\, l \in \widetilde{\mathcal{L}}, \\
& \sum_{i=1}^{M} A_{li}x_i \le C_l \qquad \forall\, l \in \mathcal{L}\setminus\widetilde{\mathcal{L}}, \\
& \sum_{l\in\widetilde{\mathcal{L}}} I_l y_l \le B, \\
& x_i \ge 0, \qquad \forall\, i = 1,\dots,M, \\
& y_l \in \{0,1\} \qquad \forall\, l \in \widetilde{\mathcal{L}}.
\end{aligned}$$

4 Numerical Tests

This section is devoted to some preliminary numerical experiments on two test networks. The numerical computation of the solutions of the GNEPs was performed by using Matlab 2018a and its optimization toolbox.

Example 1 We consider the network shown in Fig. 1 (see also [1]) with nine nodes and nine links. The origin-destination pairs of the users and their routes are described in Table 1.

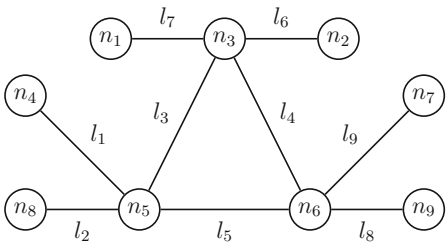


Fig. 1 Network topology of Example 1

Table 1 Origin-Destination pairs and routes (sequence of links) of the users in Example 1

User	Origin	Destination	Route
1	n_8	n_2	l_2, l_3, l_6
2	n_8	n_7	l_2, l_5, l_9
3	n_4	n_7	l_1, l_5, l_9
4	n_2	n_7	l_6, l_4, l_9
5	n_9	n_7	l_8, l_9

First, we show three instances of the considered GNEP where the variational equilibrium (1) belongs to the interior of the feasible region X; (2) has some components equal to 0; (3) saturates the capacity constraint of some links:

(1) If we set parameters $e = 0.01$, $k = 1$, $u_i = 10$ for any $i = 1, \ldots, 5$, and $C_l = 10$ for any $l \in \mathcal{L}$, then the variational equilibrium is $\bar{x} = (6.70,\ 2.02,\ 2.66,\ 2.03,\ 2.70)$ with the corresponding link flow equal to $(2.66,\ 8.72,\ 6.70,\ 2.03,\ 4.67,\ 8.73,\ 0,\ 2.70,\ 9.40)$, hence $\bar{x}$ belongs to the interior of X.
(2) If we set $e = 0.01, k = 1, u_1 = 0.01, u_i = 10$ for any $i = 2, \ldots, 5$ and $C_l = 10$ for any $l \in \mathcal{L}$, then the variational equilibrium is $(0,\ 2.34,\ 2.34,\ 2.36,\ 2.38)$.
(3) If we set $e = 0.1$, $k = 0.01$, $u_i = 100$ for any $i = 1, \ldots, 5$, and $C_l = 10$ for any $l \in \mathcal{L}$, then the variational equilibrium is $(7.8429,\ 2.1571,\ 2.8429,\ 2.1571,\ 2.8430)$ and the corresponding link flow is equal to $(2.84, 10, 7.84, 2.16, 5.00, 10, 0, 2.84, 10)$, where the capacity constraints of links l_2, l_6 and l_9 are saturated.

Moreover, we notice that there can be (infinitely) many generalized Nash equilibria unlike the unique variational one. For example, in the instance (3) a non-variational equilibrium is $(8.0944, 1.9056, 3.0944, 1.9056, 3.0944)$. It is a so-called normalized equilibrium [12], which has been computed by solving the variational inequality $VI(F', X)$, where $F_i' = w_i F_i$, for $i = 1, \ldots, 5$, and the vector of weights $w = (1/3, 1/6, 1/6, 1/6, 1/6)$ (see [10]). Similarly, other (normalized) generalized Nash equilibria can be obtained by appropriately modifying the vector w.

We now show some numerical results for the proposed optimal network improvement model. We set $e = 0.01$, $k = 1$, $u_i = 10$ for any $i = 1, \ldots, 5$, and $C_l = 10$ for any $l \in \mathcal{L}$. We assume that the available budget $B = 20$ k€, the set of links to be maintained is $\widetilde{\mathcal{L}} = \mathcal{L}$, while the values of γ_l and I_l are shown in Table 2.

Table 3 shows the ten best feasible solutions together with the percentage of total cost improvement $\varphi(y)$ and the corresponding investment $I(y) = \sum_{l \in \widetilde{\mathcal{L}}} I_l y_l$. It is interesting noting that the fifth to tenth solutions have very similar values but the tenth one needs a much lower investment than the others.

Table 2 Capacity enhancement factors and investments for links of Example 1

Links	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9
γ_l	1.2	1.5	1.1	1.6	1.3	1.4	1.1	1.7	1.3
I_l (k€)	3	8	2	10	4	5	2	12	4

Table 3 The ten best feasible solutions for the optimal network improvement model in Example 1

Ranking	y	$\varphi(y)$	$I(y)$
1	(0,1,1,0,0,1,0,0,1)	17.7381	19
2	(1,1,0,0,0,1,0,0,1)	16.8796	20
3	(0,1,0,0,0,1,1,0,1)	16.8426	19
4	(0,1,0,0,0,1,0,0,1)	16.8285	17
5	(0,1,1,0,1,0,1,0,1)	13.2289	20
6	(0,1,1,0,1,0,0,0,1)	13.2148	18
7	(1,0,1,0,1,1,1,0,1)	13.1952	20
8	(1,0,1,0,1,1,0,0,1)	13.1811	18
9	(0,0,1,0,1,1,1,0,1)	13.1396	17
10	(0,0,1,0,1,1,0,0,1)	13.1255	15

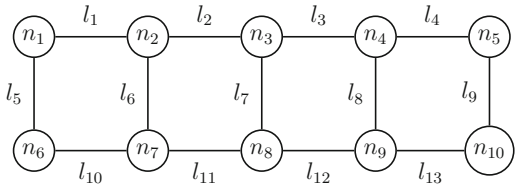


Fig. 2 Network topology of Example 2

Table 4 Origin-Destination pairs and routes (sequence of links) of the users in Example 2

User	Origin	Destination	Route	User	Origin	Destination	Route
1	n_1	n_5	l_1, l_2, l_3, l_4	6	n_5	n_1	l_4, l_3, l_2, l_1
2	n_6	n_{10}	$l_{10}, l_{11}, l_{12}, l_{13}$	7	n_{10}	n_6	$l_{13}, l_{12}, l_{11}, l_{10}$
3	n_2	n_{10}	$l_6, l_{11}, l_{12}, l_{13}$	8	n_5	n_8	l_9, l_{13}, l_{12}
4	n_8	n_5	l_7, l_3, l_4	9	n_4	n_6	$l_8, l_{12}, l_{11}, l_{10}$
5	n_6	n_5	l_5, l_1, l_2, l_3, l_4	10	n_8	n_1	l_7, l_2, l_1

Table 5 Capacity enhancement factors and investments for links of Example 2

Links	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}	l_{11}	l_{12}	l_{13}
γ_l	1.2	1.5	1.1	1.6	1.3	1.4	1.1	1.7	1.3	1.5	1.1	1.8	1.3
I_l (k€)	3	8	2	10	4	5	2	12	4	8	2	13	4

Example 2 We now consider the network shown in Fig. 2 with 10 nodes and 13 links. The O-D pairs of the users and their routes are described in Table 4.

We now show some numerical results for the proposed optimal network improvement model. We set $e = 0.01$, $k = 1$, $u_i = 10$ for any $i = 1, \ldots, 10$, and $C_l = 10$ for any $l \in \mathcal{L}$. We assume that the available budget $B = 20$ k€, the set of links to be maintained is $\widetilde{\mathcal{L}} = \mathcal{L}$, while the values of γ_l and I_l are shown in Table 5.

Table 6 shows the ten best feasible solutions together with the value of φ and the corresponding investment I.

Table 6 The ten best feasible solutions for the optimal network improvement model in Example 2

Ranking	y	$\varphi(y)$	$I(y)$
1	(0,0,0,0,0,0,0,0,0,0,1,1,1)	14.2823	19
2	(1,0,0,0,0,0,0,0,0,0,0,1,1)	13.3226	20
3	(0,0,1,0,0,0,0,0,0,0,0,1,1)	13.1370	19
4	(0,0,0,0,0,0,1,0,0,0,0,1,1)	12.7595	19
5	(0,0,0,0,0,0,0,0,0,0,0,1,1)	12.6185	17
6	(1,0,1,0,0,0,0,0,0,0,1,1,0)	11.0705	20
7	(1,0,0,0,0,0,1,0,0,0,1,1,0)	10.6949	20
8	(1,0,0,0,0,0,0,0,0,0,1,1,0)	10.5456	18
9	(0,0,1,0,0,0,1,0,0,0,1,1,0)	10.5079	19
10	(0,0,1,0,0,0,0,0,0,0,1,1,0)	10.3600	17

5 Conclusions and Future Directions

In this note we investigated a game theoretic model of congestion control in communication networks which is widely used in the literature on this topic. After introducing some modifications in the model, we studied an investment optimization problem that the network system manager faces in order to improve the capacity of links.

An interesting extension of this model could be considering the possibility that some of the data are not deterministic but random. Since we used the variational inequality approach to GNEP, the inclusion of such random data should be performed within the framework of stochastic variational inequalities (see, e.g., [6, 7, 9]).

Acknowledgments The authors are members of the Gruppo Nazionale per l'Analisi Matematica, la Probabilità e le loro Applicazioni (GNAMPA—National Group for Mathematical Analysis, Probability and their Applications) of the Istituto Nazionale di Alta Matematica (INdAM—National Institute of Higher Mathematics).

References

1. Alpcan, T., Başar, T.: A game-theoretic framework for congestion control in general topology networks. In: Proceedings of the IEEE 41st Conference on Decision and Control Las Vegas, December 10–13 (2002)
2. Altman, E. Başar, T., Jimenez, T., Shimkin, N.: Competitive routing in networks with polynomial costs. IEEE Trans. Autom. Control **47**, 92–96 (2002)
3. Facchinei, F., Fischer, A., Piccialli, V.: On generalized Nash games and variational inequalities. Oper. Res. Lett. **35**, 159–164 (2007)
4. Facchinei, F., Kanzov, C.: Generalized Nash equilibrium problems. Ann. Oper. Res. **175**, 177–211 (2010)

5. Faraci, F., Raciti, F.: On generalized Nash equilibrium problems in infinite dimension: the Lagrange multipliers approach. Optimization **64**, 321–338 (2015)
6. Jadamba, B., Pappalardo, M., Raciti, F.: Efficiency and vulnerability analysis for congested networks with random data. J. Optim. Theory Appl. **177**, 563–583 (2018)
7. Koshal, J., Nedić, A., Shanbhag, U.V.: Regularized iterative stochastic approximation methods for stochastic variational inequality problems. IEEE Trans. Autom. Control **58**, 594–609 (2013)
8. Mastroeni, G., Pappalardo M., Raciti, F.: Generalized Nash equilibrium problems and variational inequalities in Lebesgue spaces. Minimax Theory Appl. **5**, 47–64 (2020)
9. Passacantando, M., Raciti, F.: Optimal road maintenance investment in traffic networks with random demands. Optim. Lett. **15**, 1799–1819 (2019)
10. Nabetani, K., Tseng, P., Fukushima, M.: Parametrized variational inequality approaches to-generalized Nash equilibrium problems with shared constraints. Comput. Optim. Appl. **48**, 423–452 (2011)
11. Orda, A., Rom, R., Shimkin, N.: Competitive routing in multiuser communication networks. IEEE/ACM Trans. Netw. **1**, 510–521 (1993)
12. Rosen, J.B.: Existence and uniqueness of equilibrium points for concave n-person games. Econometrica **33**, 520–534 (1965)
13. Yin, H., Shanbhag, U.V., Metha, P.G.: Nash equilibrium problems with congestion costs and shared constraints. In: Proceedings of the 48h IEEE Conference on Decision and Control held jointly with 2009 28th Chinese Control Conference, Shanghai, December 16–18 (2009)

A Note on Network Games with Strategic Complements and the Katz-Bonacich Centrality Measure

Mauro Passacantando and Fabio Raciti

Abstract We investigate a class of network games with strategic complements and bounded strategy sets by using the variational inequality approach. In the case where the Nash equilibrium of the game has some boundary components, we derive a formula which connects the equilibrium to the Katz-Bonacich centrality measure, thus generalizing the classical result for the interior solution case. Furthermore, we prove that any component of the Nash equilibrium is less than or equal to the corresponding component of the social optimal solution and numerically study the price of anarchy for a small size test problem.

Keywords Network games · Nash equilibrium · Katz-Bonacich centrality measure · Price of anarchy

1 Introduction

This paper deals with a class of Network Games (see, e.g., [10]), where each player is identified with the node of a graph and players that can interact directly are connected through edges of the graph. This kind of games have proven to be very useful in modeling social and economic interactions, where the action of a typical player is likely to be influenced by the actions taken by her/his friends or colleagues. A feature of these models is the central role played by the graph structure in influencing the social or economic interactions and shaping the resulting equilibrium. As a consequence, the Nash equilibrium and the social optimal solution depend on graph-algebraic quantities. In particular, in the case of interior solution

M. Passacantando
Department of Computer Science, University of Pisa, Pisa, Italy
e-mail: mauro.passacantando@unipi.it

F. Raciti (✉)
Department of Mathematics and Computer Science, University of Catania, Catania, Italy
e-mail: fraciti@dmi.unict.it

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_5

a very interesting representation formula has been derived in the seminal paper by Ballester et al. [1], which involves the so called Katz-Bonacich centrality measure.

In this note, we generalize this formula to the case where the solution has some boundary components. To this end, we focus on the quadratic reference model with strategic complements which, roughly speaking, describe social interactions where the incentive for a player to take an action increases when the number of her/his social contacts who take the action increases. Furthermore, by exploiting the sequential best-response dynamics, we prove that the Nash equilibrium is component-wise less than or equal to the social optimal solution. To obtain our results, we reformulate our problem as an equivalent variational inequality. The relationship between Nash equilibrium problems and variational inequalities has been pioneered by Gabay and Moulin [8], but only recently some authors have applied this methodology to investigate Network Games. In these regards, we refer the interested reader to the beautiful paper by Parise and Ozdaglar [15], which although comprehensive in many respects such as uniqueness and sensitivity of equilibrium, does not focus on the Katz-Bonacich representation of the solution or on the price of anarchy.

The paper is structured as follows. In Sect. 2 we first introduce the notation, the basic definitions and the variational inequality approach. We then specialize to the reference quadratic model and recall the classical Katz-Bonacich formula for the interior solution case, where the strategy set of each player is $\mathbb{R}_+$. We also recall the notions of social optimal solution, efficiency of a Nash equilibrium, and price of anarchy. In Sect. 3, we assume that the strategy sets are bounded also from above and derive a necessary condition that the solution satisfies when some of its components lie on the boundary (Theorem 3). We interpret this condition in terms of the Katz-Bonacich centrality measure. Moreover, we prove the relationship between the Nash equilibrium and the social optimum (Theorem 4). Theorems 3 and 4 of this section are, to the best of our knowledge, new and represent the main contribution of this note, Sect. 4 is devoted to the numerical investigation of a test problem to illustrate our findings. A short concluding section ends the paper.

2 Network Games

2.1 Game Formulation and Variational Inequality Approach

In Network Games players are represented by the nodes of an undirected graph (V, E), where $V = \{v_1, \ldots, v_n\}$ is the sets of nodes and E is the set of edges formed by pairs of nodes (v_i, v_j). Here, we consider undirected simple graphs. Two nodes v_i and v_j are said to be adjacent if they are connected by the edge (v_i, v_j). The information about the adjacency of nodes can be stored in the adjacency matrix G whose elements g_{ij} are equal to 1 if (v_i, v_j) is an edge, 0 otherwise. G is thus a symmetric and zero-diagonal matrix. Given a node v, the nodes connected to v

with an edge are called the *neighbors* of v. A *walk* in the graph is a finite sequence of the form $v_{i_0}, e_{j_1}, v_{i_1}, e_{j_2}, \ldots, e_{j_k}, v_{j_k}$, which consists of alternating nodes and edges of the graph, such that $v_{i_{t-1}}$ and v_{i_t} are end nodes of edge e_{j_t}. The *length* of a walk is the number of its edges. Let us remark that it is allowed to visit a node or go through an edge more than once. The indirect connections between any two nodes in the graph are described by means of the powers of the adjacency matrix G. Indeed, it can be proved that the element $g_{ij}^{[k]}$ of G^k gives the number of walks of length k between nodes v_i and v_j.

In the sequel, the set of players will be denoted by $\{1, 2, \ldots, n\}$ instead of $\{v_1, v_2, \ldots, v_n\}$. We denote with $A_i \subset \mathbb{R}$ the action space of player i, while $A = A_1 \times \cdots \times A_n$. For each $a = (a_1, \ldots, a_n)$, $a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$ and the notation $a = (a_i, a_{-i})$ will be used when we want to distinguish the action of player i from the action of all the other players. Each player i is endowed with a payoff function $u_i : A \to \mathbb{R}$ that she/he wishes to maximize. The notation $u_i(a, G)$ is often utilized when one wants to emphasize that the utility of player i also depends on the actions taken by her/his neighbors in the graph.

The solution concept that we consider here is the Nash equilibrium of the game, that is, we seek an element $a^* \in A$ such that for each $i \in \{1, \ldots, n\}$:

$$u_i(a_i^*, a_{-i}^*) \geq u_i(a_i, a_{-i}^*), \qquad \forall\, a_i \in A_i. \tag{1}$$

We now posit a further assumption on how variations of the actions of player i's neighbors affect her/his marginal utility.

Definition 1 The network game has the property of strategic complements if:

$$\frac{\partial^2 u_i}{\partial a_j \partial a_i}(a) > 0, \qquad \forall\, (i, j) :\ g_{ij} = 1,\ \forall\, a \in A.$$

For the subsequent development it is important to recall that if the u_i are continuously differentiable functions on A, the Nash equilibrium problem is equivalent to the variational inequality $VI(F, A)$: find $a^* \in A$ such that

$$[F(a^*)]^\top (a - a^*) \geq 0, \qquad \forall\, a \in A, \tag{2}$$

where

$$[F(a)]^\top := -\left(\frac{\partial u_1}{\partial a_1}(a), \ldots, \frac{\partial u_n}{\partial a_n}(a)\right) \tag{3}$$

is also called the pseudo-gradient of the game. For an account of variational inequalities the interested reader can refer to [7, 12, 14]. We recall here some useful monotonicity properties.

Definition 2 A map $T : \mathbb{R}^n \to \mathbb{R}^n$ is said to be monotone on A iff:

$$[T(x) - T(y)]^\top (x - y) \geq 0, \qquad \forall\, x, y \in A.$$

If the equality holds only when $x = y$, T is said to be strictly monotone.

T is said to be β-strongly monotone on A iff there exists $\beta > 0$ such that

$$[T(x) - T(y)]^\top (x - y) \geq \beta \|x - y\|^2, \qquad \forall\, x, y \in A.$$

For linear operators on $\mathbb{R}^n$ the two concepts of strict and strong monotonicity coincide and are equivalent to the positive definiteness of the corresponding matrix.

Conditions that ensure the unique solvability of a variational inequality problem are given by the following theorem (see, e.g., [7, 14]).

Theorem 1 *If $K \subset \mathbb{R}^n$ is a compact convex set and $T : \mathbb{R}^n \to \mathbb{R}^n$ is continuous on K, then the variational inequality problem $VI(F, K)$ admits at least one solution. In the case K is unbounded, existence of a solution may be established under the following coercivity condition:*

$$\lim_{\|x\| \to +\infty} \frac{[T(x) - T(x_0)]^\top (x - x_0)}{\|x - x_0\|} = +\infty,$$

for $x \in K$ and some $x_0 \in K$.

Furthermore, if T is strictly monotone on K the solution is unique.

In the following subsection, we describe in detail the linear-quadratic reference model with strategic complements.

2.2 The Linear-Quadratic Model

Let $A_i = \mathbb{R}_+$ for any $i \in \{1, \ldots, n\}$, hence $A = \mathbb{R}^n_+$. The payoff of player i is given by:

$$u_i(a, G) = -\frac{1}{2} a_i^2 + \alpha a_i + \phi \sum_{j=1}^{n} g_{ij} a_i a_j, \qquad \alpha, \phi > 0. \tag{4}$$

In this simplified model α and ϕ take on the same value for all players, which then only differ according to their position in the network. The last term describes the interaction between neighbors, and since $\phi > 0$ this interaction falls in the class of strategic complements. The pseudo-gradient's components of this game are easily computed as:

$$F_i(a) = a_i - \alpha - \phi \sum_{j=1}^{n} g_{ij} a_j, \qquad i \in \{1, \ldots, n\},$$

which can be written in compact form as $F(a) = (I - \phi G)a - \alpha \mathbf{1}$, where $\mathbf{1} = (1, \ldots, 1)^\top \in \mathbb{R}^n$. We will seek Nash equilibrium points by solving the variational inequality:

$$[F(a^*)]^\top (a - a^*) \geq 0, \qquad \forall\, a \in \mathbb{R}^n_+. \tag{5}$$

Since the constraint set is unbounded, to ensure solvability we require that F be strongly monotone, which (implying coercivity, for linear operators) also guarantees the uniqueness of the solution.

Lemma 1 (see, e.g. [10]) *The matrix $I - \phi G$ is positive definite iff $\phi\rho(G) < 1$, where $\rho(G)$ is the spectral radius of G.*

In the next lemma we recall a well known result about series of matrices.

Lemma 2 *Let T be a square matrix and consider the series $\sum_{p=0}^{\infty} T^p$. The series converges provided that $\lim_{p\to\infty} T^p = 0$, which is equivalent to $\rho(T) < 1$. In such case the matrix $I - T$ is non singular and we have $(I - T)^{-1} = \sum_{p=0}^{\infty} T^p$.*

Theorem 2 (See e.g. [10]) *If $\phi\rho(G) < 1$, then the unique Nash equilibrium is*

$$a^* = \alpha(I - \phi G)^{-1}\mathbf{1} = \alpha \sum_{p=0}^{\infty} \phi^p G^p \mathbf{1}\,. \tag{6}$$

Remark 1 The expansion in (6) suggests an interesting interpretation. Indeed, the (i, j) entry, $g_{ij}^{[p]}$, of the matrix G^p gives the number of walks of length p between nodes i and j. Based on this observation, a measure of centrality on the network was proposed by Katz and Bonacich [5]. Specifically, for any weight $w \in \mathbb{R}^n_+$, the weighted vector of Katz-Bonacich is given by:

$$b_w(G, \phi) = M(G, \phi)w = (I - \phi G)^{-1} w = \sum_{p=0}^{\infty} \phi^p G^p w. \tag{7}$$

In the case where $w = \mathbf{1}$, the (non weighted) centrality measure of Katz-Bonacich of node i is given by:

$$b_{1,i}(G, \phi) = \sum_{j=1}^{n} M_{ij}(G, \phi)$$

and counts the total number of walks in the graph, which start at node i, exponentially damped by ϕ.

Remark 2 The game under consideration also falls in the class of potential games according to the definition introduced by Monderer and Shapley [13]. Indeed, a potential function is given by:

$$P(a, G, \phi) = \sum_{i=1}^{n} u_i(a, G) - \frac{\phi}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} g_{ij} a_i a_j.$$

Monderer and Shapley have proved that, in general, the solutions of the problem $\max_{a \in A} P(a, G, \phi)$ form a subset of the solution set of the Nash game. Because under the condition $\phi\rho(G) < 1$ both problems have a unique solution, it follows that the two problems share the same solution.

Let us now define the social welfare function as:

$$W(a, G) = \sum_{i=1}^{n} u_i(a, G) = -\frac{1}{2} a^\top (I - 2\phi G)a + \alpha \mathbf{1}^\top a.$$

The value that W takes on at the Nash equilibrium a^* can be easily computed as $W(a^*, G) = \frac{1}{2}\alpha^2 b_{\mathbf{1}}(G, \phi)^\top b_{\mathbf{1}}(G, \phi)$. The natural question arises of comparing this value with the optimal value of W. Under the condition $2\phi\rho(G) < 1$ it turns out that the maximum of W is reached at $a^{so} = \alpha b_{\mathbf{1}}(G, 2\phi)$ and $W(a^{so}, G) = \frac{1}{2}\alpha^2 b_{\mathbf{1}}(G, 2\phi)^\top b_{\mathbf{1}}(G, 2\phi)$. Thus, the Nash equilibrium is not efficient and it is interesting to compute the ratio:

$$\gamma(G, \phi) = \frac{W(a^*, G)}{W(a^{so}, G)}, \tag{8}$$

which can be termed the *price of anarchy* (see, e.g., [16]).

3 Bounded Strategies

We now assume that the strategies of each player have an upper bound, i.e., the strategy set $A_i = [0, L_i]$, with $L_i > 0$, for any $i \in \{1, \ldots, n\}$, and derive a Katz-Bonacich type representation of the solution, in the case where exactly k components take on their maximum value.

Theorem 3 *Let u_i be defined as in (4), $\phi\rho(G) < 1$, $A_i = [0, L_i]$ for any $i \in \{1, \ldots, n\}$ and a^* be the unique Nash equilibrium of the game. We then have that $a_i^* > 0$ for any $i \in \{1, \ldots, n\}$. Moreover, assume that exactly k components of a^* take on their maximum value: $a_{i_1}^* = L_{i_1}, \ldots, a_{i_k}^* = L_{i_k}$, and denote with $\tilde{a}^* = (\tilde{a}_{i_{k+1}}^*, \ldots, \tilde{a}_{i_n}^*)$ the subvector of the nonboundary components of a^*. We then get:*

$$\tilde{a}^* = (I_{n-k} - \phi\, G_1)^{-1} w = b_w(G_1, \phi), \tag{9}$$

where G_1 is the submatrix obtained from G choosing the rows $i_{k+1}, \ldots, i_n$ and the columns $i_{k+1}, \ldots, i_n$; G_2 is the submatrix obtained from G choosing the rows $i_{k+1}, \ldots, i_n$ and the columns $i_1, \ldots, i_k$; $w = \alpha \mathbf{1}_{n-k} + \phi G_2 L$ and $L = (L_{i_1}, \ldots, L_{i_k})$.

Proof The Nash equilibrium $a^*$of the game solves the variational inequality

$$[F(a^*)]^\top (a - a^*) \geq 0, \qquad \forall\, a \in A, \tag{10}$$

where $A = [0, L_1] \times \cdots \times [0, L_n]$. Let us assume that there exist l such that $a_l^* = 0$, and choose in (10) $a = (a_1^*, \ldots, a_{l-1}^*, L_l, a_{l+1}^*, \ldots, a_n^*) \in A$. With this choice, (10) reads:

$$0 \leq F_l(a^*) L_l = \left(-\phi \sum_{j=1}^{n} g_{lj} a_j^* - \alpha \right) L_l < 0,$$

which yields the contradiction. Thus, $a_i^* > 0$ for any $i = 1, \ldots, n$.

Let $\tilde{A}$ denote the face of A obtained intersecting A with the hyperplanes: $a_{i_1} = L_{i_1}, \ldots, a_{i_k} = L_{i_k}$. Moreover, let $\tilde{a} = (a_{i_{k+1}}, \ldots, a_{i_n})$, $\tilde{a}^* = (\tilde{a}_{i_{k+1}}^*, \ldots, \tilde{a}_{i_n}^*)$ and $\tilde{F} : \mathbb{R}^{n-k} \to \mathbb{R}^{n-k}$ such that $\tilde{F}_{i_l}(\tilde{a})$ is obtained by fixing $a_{i_1} = L_{i_1}, \ldots, a_{i_k} = L_{i_k}$ in $F_{i_l}(a)$. We consider now the restriction of (10) to $\tilde{A}$, which reads:

$$\sum_{l=k+1}^{n} \tilde{F}_{i_l}(\tilde{a}^*)(\tilde{a}_{i_l} - \tilde{a}_{i_l}^*) \geq 0, \qquad \forall\, \tilde{a} \in \tilde{A}. \tag{11}$$

Since we are assuming that exactly k components of the solution a^* reach their upper bounds, it follows that $\tilde{a}^*$ lies in the interior of $\tilde{A}$, hence $\tilde{F}(\tilde{a}^*) = 0$, that is equivalent to

$$a_{i_l}^* - \phi \sum_{m=k+1}^{n} g_{i_l i_m} a_{i_m}^* = \alpha + \phi \sum_{m=1}^{k} g_{i_l i_m} L_{i_m}, \quad l = k+1, \ldots, n, \tag{12}$$

which yields:

$$(I_{n-k} - \phi G_1)\tilde{a}^* = \alpha \mathbf{1}_{n-k} + \phi G_2 L. \tag{13}$$

Because the matrix $(I_{n-k} - \phi G_1)$ is not singular, the thesis is proved. □

The following result shows a relationship between the Nash equilibrium and the social optimum of the game.

Theorem 4 *Let u_i be defined as in (4), $\phi\rho(G) < 1/2$, and $A_i = [0, L_i]$ for any $i \in \{1, \dots, n\}$. Then,*

$$a_i^* \le a_i^{so} \qquad \forall\, i = 1, \dots, n,$$

where a^* is the Nash equilibrium and a^{so} is the social optimum of the game.

Proof Since $\phi\rho(G) < 1/2$, there exists a unique Nash equilibrium a^* and a unique social optimum a^{so}. Moreover, it follows from the KKT conditions that a^{so} satisfies the following system: $a_i^{so} = \min\left\{L_i,\ \alpha + 2\phi \sum_{j=1}^n g_{ij} a_j^{so}\right\}$, $i = 1, \dots, n$. Given any strategy profile $a = (a_i, a_{-i})$, the best response of player i to rivals' strategies a_{-i} is given by $B_i(a_{-i}) = \arg\max_{a_i \in [0, L_i]} u_i(\cdot, a_{-i}) = \min\left\{L_i,\ \alpha + \phi \sum_{j=1}^n g_{ij} a_j\right\}$.

We now consider the sequential best response dynamics starting from the social optimum a^{so}, that is the sequence $\{a^k\}$ defined as follows:

$$\begin{aligned}
a^0 &= a^{so}, a^1 = \left(B_1(a_{-1}^0),\ a_2^0,\ a_3^0,\ \dots\ ,\ a_n^0\right),\\
a^2 &= \left(B_1(a_{-1}^0),\ B_2(a_{-2}^1),\ a_3^0,\ \dots\ ,\ a_n^0\right),\\
&\dots\\
a^n &= \left(B_1(a_{-1}^0),\ B_2(a_{-2}^1),\ B_3(a_{-3}^2),\ \dots\ ,\ B_n(a_{-n}^{n-1})\right),\\
a^{n+1} &= \left(B_1(a_{-1}^n),\ B_2(a_{-2}^1),\ \dots\ ,\ B_n(a_{-n}^{n-1})\right),\\
a^{n+2} &= \left(B_1(a_{-1}^n),\ B_2(a_{-2}^{n+1}),\ B_3(a_{-3}^2),\ \dots\ ,\ B_n(a_{-n}^{n-1})\right), \dots .
\end{aligned}$$

We note that

$$a_1^1 = B_1\left(a_{-1}^0\right) = \min\left\{L_1,\ \alpha + \phi \sum_{j=1}^n g_{1j} a_j^0\right\} \le \min\left\{L_1,\ \alpha + 2\phi \sum_{j=1}^n g_{1j} a_j^0\right\} = a_1^0,$$

hence $a^1 \le a^0$. Moreover, we have

$$a_2^2 = \min\left\{L_2,\ \alpha + \phi \sum_{j=1}^n g_{2j} a_j^1\right\} \le \min\left\{L_2,\ \alpha + 2\phi \sum_{j=1}^n g_{2j} a_j^0\right\} = a_2^0 = a_2^1,$$

hence $a^2 \leq a^1$. Similarly, we can prove that $a^n \leq a^{n-1} \leq \cdots \leq a^1 \leq a^0$. Furthermore, we get

$$a_1^{n+1} = \min\left\{L_1,\ \alpha + \phi \sum_{j=1}^{n} g_{1j} a_j^n\right\} \leq \min\left\{L_1,\ \alpha + \phi \sum_{j=1}^{n} g_{1j} a_j^0\right\} = B_1\left(a_{-1}^0\right) = a_1^n,$$

hence $a^{n+1} \leq a^n$, and

$$a_2^{n+2} = \min\left\{L_2,\ \alpha + \phi \sum_{j=1}^{n} g_{2j} a_j^{n+1}\right\} \leq \min\left\{L_2,\ \alpha + \phi \sum_{j=1}^{n} g_{2j} a_j^1\right\} = a_2^{n+1},$$

thus $a^{n+2} \leq a^{n+1}$. Following the same argument as before, we can prove that $a^{k+1} \leq a^k$ for any $k \in \mathbb{N}$ and hence, in particular, $a^k \leq a^{so}$ holds for any k. Since the potential function P is strongly concave, the sequence $\{a^k\}$ converges to the unique Nash equilibrium a^* (see, e.g., [4, Proposition 3.9]), hence $a^* \leq a^{so}$. □

4 Numerical Example

In this section, we show a numerical example for the linear-quadratic network game described in Sect. 3.

Example 1 We consider the network shown in Fig. 1 (see also [3]) with eight nodes (players). The spectral radius of the adjacency matrix G is $\rho(G) \simeq 3.1019$. We set parameters $\alpha = 10$, $\phi = 0.45/\rho(G)$ and upper bounds $L_i = L = 18$ for any player $i = 1, \ldots, 8$. Therefore, there exists a unique Nash equilibrium and a unique social optimum. Table 1 shows the unconstrained Nash equilibrium (assuming $L = +\infty$, given by formula (6)), the constrained Nash equilibrium (assuming $L = 18$) and the social optimum.

Figure 2 shows the price of anarchy $\gamma(G, \phi)$ of the Nash equilibrium for different values of L and ϕ. The results suggest that (1) the price of anarchy is a non-increasing function of L; (2) it is constant when either L is small enough (i.e., the Nash equilibrium coincides with the social optimum) or greater than some threshold

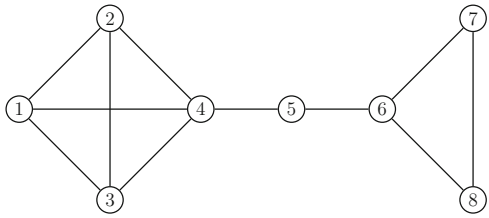


Fig. 1 Network topology of Example 1

Table 1 Unconstrained Nash equilibrium, constrained Nash equilibrium (considering upper bound $L = 18$) and social optimum for Example 1

Player	Unconstrained NE	Constrained NE	Social optimum
1	18.2041	17.7661	18.0000
2	18.2041	17.7661	18.0000
3	18.2041	17.7661	18.0000
4	20.1431	18.0000	18.0000
5	15.3047	14.9868	18.0000
6	16.4227	16.3742	18.0000
7	14.4837	14.4755	18.0000
8	14.4837	14.4755	18.0000

Fig. 2 Price of anarchy for different values of L and ϕ

(i.e., the Nash equilibrium and the social optimum are both interior to the feasible region); (3) the larger the value of ϕ, the smaller the asymptotic value of $\gamma(G, \phi)$ is.

5 Conclusions and Further Research Perspectives

A future research direction is the use of the necessary condition for boundary solutions to develop an algorithm for finding the Nash equilibrium. Moreover, the introduction of uncertain data in the model could be done along the same lines as in [11]. From the application viewpoint, our results could be used to further develop

several models in social sciences and economy. For instance, in [2] criminal social interactions were analyzed within the framework of a general quadratic model in social networks; in [6], the influence of peers on educational networks has been studied extensively using the approach described in this note; moreover, in [9] the quadratic model was used to investigate the interaction between the social space (i.e., the network) and the geographical space (i.e., the city).

Acknowledgments The authors are members of the Gruppo Nazionale per l'Analisi Matematica, la Probabilità e le loro Applicazioni (GNAMPA—National Group for Mathematical Analysis, Probability and their Applications) of the Istituto Nazionale di Alta Matematica (INdAM—National Institute of Higher Mathematics).

References

1. Ballester, C., Calvo-Armengol, A., Zenou, Y.: Who's who in networks. Wanted: the key player. Econometrica **74**, 1403–1417 (2006)
2. Ballester, C., Calvo-Armengol, A., Zenou, Y.: Delinquent networks. J. Eur. Econ. Assoc. **8**, 34–61 (2010)
3. Belhaj, M., Bramoulleé, Y., Deroian, F.: Network games under strategic complementarities. Games Econon. Behav. **88**, 310–319 (2014)
4. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and Distributed Computation: Numerical Methods. Athena Scientific (1997)
5. Bonacich, P.: Power and centrality: a family of measures. Am. J. Sociol. **92**, 1170–1182 (1987)
6. Calvo-Armengol, A., Patacchini, E., Zenou, Y.: Peer effects and social networks in education. Rev. Econ. Stud. **76**, 1239–1267 (2009)
7. Facchinei, F., Pang, J.-S.: Finite-Dimensional Variational Inequalities and Complementarity Problems. Springer, Berlin (2003)
8. Gabay, D., Moulin, H.: On the uniqueness and stability of Nash equilibria in noncooperative games. In: Bensoussan, A., Kleindorfer, P., Tapiero, C.S. (eds) Applied Stochastic Control in Econometrics and Management Science. North-Holland, Amsterdam, pp. 271–294 (1980)
9. Helsley, R. Zenou, Y.: Social networks and interactions in cities. J. Econ. Theory **150**, 426–466 (2014)
10. Jackson, M.O., Zenou, Y.: Games on networks. In: Handbook of Game Theory with Economic Applications, pp. 95–163. Elsevier, Amsterdam (2015)
11. Jadamba, B., Raciti, F.: On the modelling of some environmental games with uncertain data. J. Optim. Theory Appl. **167**, 959–968 (2015)
12. Konnov, I.: Equilibrium Models and Variational Inequalities. Elsevier, Amsterdam (2007)
13. Monderer, D., Shapley, L.S.: Potential games. Games Econ. Behav. **14**, 124–143 (1996)
14. Nagurney, A.: Network Economics: A Variational Inequality Approach. Springer, New York (1999)
15. Parise, F. Ozdaglar, A.: A variational inequality framework for network games: Existence, uniqueness, convergence and sensitivity analysis. Games Econ. Behav. **114**, 47–82 (2019)
16. Roughgarden, T., Tardos, E.: Bounding the inefficiency of equilibria in nonatomic congestion games. Games Econ. Behav. **47**, 389–403 (2004)

Part II
Healthcare

An Optimization Model for Managing Reagents and Swab Testing During the COVID-19 Pandemic

Gabriella Colajanni, Patrizia Daniele, and Veronica Biazzo

Abstract The ongoing COVID-19 pandemic, caused by Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2), is having an irreversible effect on millions of people around the world. This new pathology is characterized by symptomatic patients who need hospital care, paucisymptomatic patients but also asymptomatic patients who could considerably spread the virus without being aware of it; therefore, the virus spreads very quickly and swab tests for viral presence are used to diagnose positive cases. In this paper we present a multi-period resource allocation model with the objective of simultaneously maximize the quantity of all analyzed swabs while minimizing the time required to obtain the swabs result, the costs due to increase the number of swabs analyzed per unit time and the cost to transfer swabs between laboratories (when a laboratory receives more swab tests than it can analyze).

Keywords COVID-19 · Multi-period model · Allocation problem

1 Introduction

In this paper we consider a model for emergency resource allocation in the event of a viral pandemic with the main goal of minimizing the impact of such an event. A severe viral pandemic or epidemic, particularly one with an high rate of contagion, could have an extremely devastating impact on million people worldwide.

First reported in the city of Wuhan, China, severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) is a novel coronavirus that has spread rapidly worldwide, leading the World Health Organization to declare a global pandemic

G. Colajanni (✉) · P. Daniele · V. Biazzo
University of Catania, Catania, Italy
e-mail: colajanni@dmi.unict.it; daniele@dmi.unict.it; vbiazzo@dmi.unict.it

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_6

of COVID-19 on the 11th of March. This rapid and extensive spread of infection and the increasing pressure on hospital capacity have led to an official lockdown in a lot of countries. These restrictive measures have generated, on the one side, a slowdown in the spread of the infection, but on the other a considerable cost to the economy (see [1]). Therefore, after many countries have lifted lockdown measures and restrictions, it is necessary to avoid overloading the health service and the rapid reappearance of infections that could cause further deaths. Moreover, although some patients require intensive care, others have mild symptoms, but many others may be asymptomatic (it is estimated they are 17.9%, see [10]). Thus, testing as many people as possible is essential in order to limit the growth of infections (see [5]) identifying positive patients. In addition, the data show that there is a correlation between the number of tested swabs and the number of new cases of COVID-19 positive people.

There exist different types of swab-tests, media, and kits can be used for COVID-19 sample collection. There are currently two primary types of COVID-19 tests: *blood tests* (or serology tests) that hunt for antibodies and *diagnostic tests* that look for active coronavirus infection in mucus or saliva using a swab (which is inserted into the throat and the nose). Some swab-tests, the *antigen tests*, look for a piece of the coating of the virus (also called rapid tests, they are generally quick and cheap but they are more prone to false negative results), and other tests, the *Molecular tests*, detect nucleic acid (such as RNA) belonging to the coronavirus. RNA tests are considered to be the ones that produce the most sensitive and highly accurate results and they are often called RT-PCR tests, short for Real Time-Polymerase Chain Reaction, the lab technique used to detect the virus's genetic material. Running a PCR test and reading its results requires specific equipment and chemicals (known as reagents) that are often in short supply because of the high number of requests.

The situation of Covid-19 has been studied in the literature, also in the field of optimization, but for different aspects (see, for example, [8, 9, 11, 12]). The objective of this paper is to formulate a multi-period resource allocation problem that allows us to plan where and when it is convenient to allocate reagents and whether to transfer the swabs to be analyzed from one laboratory to another one (minimizing the transfer costs). This planning model establishes the optimal number of swabs that each laboratory has to analyze in order to maximize them (while minimizing the result times), but also the amount of reagents to stock at each period and to self-produce (if the laboratory is able to do it). Moreover, this multi-period problem determines whether it is convenient for a laboratory to increase the service rate (the number of analyzed swabs per time unit), minimizing the related costs associated with it.

This paper is organized as follows. In Sect. 2 we present the mathematical model relating to a supply chain in an epidemic or pandemic situation, consisting of companies producing reagents and laboratories that analyze swabs received from local testing centers or directly from people. Therefore, we derive the optimality conditions given by the aim of maximizing the quantity of processed swabs while minimizing the delay and the costs due to the increase in the service rate and the transfer of swabs to other laboratories. In Sect. 3 we apply the model to some

numerical examples in order to emphasize the highlights of the model. Section 4 is dedicated to the conclusions.

2 The Model for Optimal Management of Reagents and Swab Testing

2.1 Problem Description

The supply chain network, consisting of companies or factories producing reagents, laboratories (hospitals, local health centers, regional health care institute, Central National Authority) which are authorized to process swabs collected from people at specialized testing sites, is depicted in Fig. 1.

Each company a, $a = 1, \ldots, A$, produces one or more types of reagents r, $r = 1, \ldots, R$, and sends them to the authorized laboratories denoted by $l = 1, \ldots, L$. Each laboratory receives reagents by productive companies (or it can self-produce them) and analyzes one or more types of swabs denoted by $s = 1, \ldots, S$. People (healthy, asymptomatic, pre-symptomatic, symptomatic) are tested in different locations (some test centers offer also a drive-through service or special care units can go to the patient's home) and then their swabs are sent to authorized laboratories or people can go directly to the laboratory to be tested (free of charge if requested by their doctor or pediatrician, for a fee otherwise). Therefore, each laboratory receives a certain amount of swabs to be analyzed.

Usually, the nearest laboratory is chosen for swab analysis, in order to reduce costs and time. In this paper, we suppose that in some cases it may be necessary to send the swabs to be analyzed to other laboratories (even further away). This can

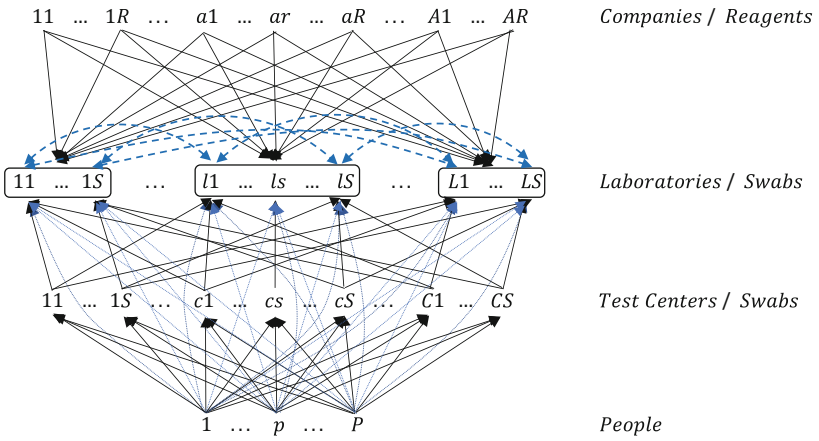


Fig. 1 Network topology

happen, for example, when, due to an epidemic outbreak, a laboratory receives too many swabs to test.

Furthermore, in this paper we analyze the model in a discrete time horizon: $1, \ldots, t, \ldots, T$.

2.2 Parameters and Variables

Now, we present the parameters of the model. Let:

- p_{rsl} be the amount of reagent r needed to process an s-type swab in the laboratory l;
- $\mathbb{E}[q_{slt}]$ be the expected value of the quantity of swabs of the type s requested to the laboratory l at time t;
- $\overline{M}_{slt}$ be the maximum number of s-type swabs that laboratory l is able to process at time t;
- $\overline{M}_{lt}$ be the maximum laboratory capacity, that is the maximum number of swabs that laboratory l is able to process at time t due to machinery or workforce;
- c_{slt} be the amount of laboratory capacity (machinery or workforce) needed to process an s-type swab in the laboratory l at time t;
- M^A_{rlt} be the maximum quantity of reagent r that the laboratory l can self-produce at time t;
- Q_{rat} be the amount of reagent r produced by and available to the company a at time t;
- $\overline{Z}_{rl}$ be the maximum number of stocks of reagent r that the laboratory l is able to maintain;
- μ^i_{sl} be the initial mean service rate, that is the number of type s swabs processed per unit of time at the laboratory l;
- ϵ_{sl} be the cost due to the additional mean service rate at the laboratory l, for swab type s;
- $\vartheta_{sl\tilde{l}}$ be the cost to transfer an s-type swab from laboratory l to $\tilde{l}$.

All variables are described in Table 1.

Table 1 Description of the variables

Variable name	Description
x_{slt}	Quantity of swabs s analyzed in l at time t
x_{ralt}	Quantity of reagent of type r sent by company a to laboratory l at time t
z_{rlt}	Quantity of reagent of type r that the laboratory l puts into storage at the end of period t
x^A_{rlt}	Quantity of type r reagent self-produced by the laboratory l at time t
μ^A_{sl}	The additional mean service rate related to the swabs s and the laboratory l
$y_{sl\tilde{l}t}$	Quantity of swabs s transferred from laboratory l to $\tilde{l}$ at time t

2.3 Objective Function

The objective function consists of four parts.

The first term maximizes the number of each type of swab analyzed in each laboratory and for each period, which is the main objective of this work. The second term aims at minimizing the delay, intended as the time required to obtain the swab result. The third element of the objective function deals with minimizing the cost due to the additional mean service rate, $\epsilon_{sl}(\mu^A_{sl})$, (when the latter is strictly positive: $\mu^A_{sl} > 0$), that is, the cost that the laboratory l has to pay to increase the number of s-type swabs analyzed per unit time (for example the cost to buy new machinery). Finally, the fourth and last part minimizes the cost to transfer swabs from a laboratory to another one, $\vartheta_{sl\tilde{l}}(y_{sl\tilde{l}t})$ (usually this happens in an outbreak situation, when a laboratory receives more swab tests than it can analyze).

As mentioned above, in the second part of the objective function we take into account the time required to obtain the swab result. Since it is independent of our variables, we do not analyze the time preceding the arrival of the swab in the laboratory, but we consider only the time from the moment the laboratory receives the swab to when the result is obtained (positive or negative for Covid). This period of time is characterized by a waiting period, due to the queue that has been generated, and by the time required to process the swab. We consider an M/M/1 queue (see [2, 6, 13]), according to the Kendall's notation (see [4]), that represents the queue length in a system in which we assume:

- having a single server (the single laboratory that receives the swabs to analyze) which serves customers one at a time from the front of the queue;
- there is only one waiting line;
- arriving requests follow each other according to a Poisson process and
- job service times have an exponential distribution.

We also assume that the buffer is of infinite size, so there is no limit on the number of customers it can contain. We choose the "first in, first out" (FIFO) Service Discipline (the swabs are analyzed in the order they arrived in).

The process describing the number of customers is Markovian. Since the probability that the chain changes from state i to state j depends only on i and j, it is an homogeneous Markov chain (see [14]).

We denote by λ the mean rate of arrivals (and, therefore, $\frac{1}{\lambda}$ represents the mean inter-arrival time) and by μ the mean service rate that is the mean number of swabs processed per unit time (and, therefore, $\frac{1}{\mu}$ represents the mean service time).

The Little's Law states that the average number of customers in a system, $\mathbb{E}[L]$, equals the average arrival rate, λ, multiplied by the average time in the system, $\mathbb{E}[S]$:

$$\mathbb{E}[L] = \lambda \mathbb{E}[S].$$

By applying the Little's law (see [7] and [3]), we obtain that the intensity of traffic, ρ, also called server utilization or occupation rate, is given by the ratio between the mean arrival rate and the mean service rate: $\rho = \frac{\lambda}{\mu}$.

Furthermore, it is clear that the average time in the system is equal to the sum between the average waiting time and the average service time: $\mathbb{E}[S] = \mathbb{E}[W] + \mathbb{E}[Z]$. Therefore, the average time in the system is given by:

$$\mathbb{E}[S] = \frac{1}{\mu - \lambda}.$$

In this model, we aim at minimizing the average time in the system for all laboratories and swab types. We consider the initial mean service rate added to the variable $\mu_{sl}^{A} \in \mathbb{R}_0^+$, the additional mean service rate, because the laboratory l could obtain an increase in the service rate, for example by purchasing additional machinery that allows to analyze more swabs per unit time (maybe even differently by swab type). Instead, the mean rate of arrivals, λ, is given by the mean of the swabs to be analyzed per unit time, that is, the expected value of the quantity of swabs requested to the laboratory from which we subtract the quantity of swabs transferred to other laboratories and we add the amount of swabs received from other laboratories.

2.4 Mathematical Modeling

The problem formulation is as follows:

$$max \sum_{s=1}^{S}\sum_{l=1}^{L}\sum_{t=1}^{T} x_{slt} - \delta \sum_{s=1}^{S}\sum_{l=1}^{L} \frac{1}{\mu_{sl}^{i} + \mu_{sl}^{A} - \left[\frac{1}{T}\sum_{t=1}^{T}\left(\mathbb{E}[q_{slt}] - \sum_{\substack{\tilde{l}\neq l \\ \tilde{l}=1}}^{L} y_{sl\tilde{l}t} + \sum_{\substack{\tilde{l}\neq l \\ \tilde{l}=1}}^{L} y_{s\tilde{l}lt}\right)\right]}$$
$$-\alpha \sum_{s=1}^{S}\sum_{l=1}^{L} \epsilon_{sl}(\mu_{sl}^{A}) - \beta \sum_{s=1}^{S}\sum_{l=1}^{L}\sum_{\substack{\tilde{l}\neq l \\ \tilde{l}=1}}^{L}\sum_{t=1}^{T} \vartheta_{sl\tilde{l}}(y_{sl\tilde{l}t}) \quad (1)$$

subject to

$$\frac{1}{T}\sum_{t=1}^{T}\left(\mathbb{E}[q_{slt}]-\sum_{\substack{\tilde{l}\neq l\\ \tilde{l}=1}}^{L} y_{sl\tilde{l}t}+\sum_{\substack{\tilde{l}\neq l\\ \tilde{l}=1}}^{L} y_{s\tilde{l}lt}\right) < \mu_{sl}^{i}+\mu_{sl}^{A} \quad \forall s=1,\ldots,S,\ \forall l=1,\ldots,L, \tag{2}$$

$$\sum_{s=1}^{S} p_{rsl}x_{slt} \leq \sum_{a=1}^{A} x_{ralt}+z_{rl(t-1)}-z_{rlt}+x_{rlt}^{A}$$
$$\forall r=1,\ldots,R,\ \forall l=1,\ldots,L,\ \forall t=1,\ldots,T, \tag{3}$$

$$x_{rlt}^{A} \leq M_{rlt}^{A} \quad \forall r=1,\ldots,R,\ \forall l=1,\ldots,L,\ \forall t=1,\ldots,T, \tag{4}$$

$$x_{slt} \leq \overline{M}_{slt}+\overline{M}_{sl}^{A}(\mu_{sl}^{A}) \quad \forall s=1,\ldots,S,\ \forall l=1,\ldots,L,\ \forall t=1,\ldots,T, \tag{5}$$

$$\sum_{s=1}^{S} c_{slt}x_{slt} \leq \overline{M}_{lt}+\sum_{s=1}^{S}\overline{M}_{sl}^{A}(\mu_{sl}^{A}) \quad \forall l=1,\ldots,L,\ \forall t=1,\ldots,T, \tag{6}$$

$$x_{slt} \leq \mathbb{E}[q_{slt}]+\sum_{\tilde{l}=1}^{L}\left(y_{s\tilde{l}lt}-y_{sl\tilde{l}t}\right) \quad \forall s=1,\ldots,S,\ \forall l=1,\ldots,L,\ \forall t=1,\ldots,T, \tag{7}$$

$$\sum_{l=1}^{L} x_{ralt} \leq Q_{rat} \quad \forall r=1,\ldots,R,\ \forall a=1,\ldots,A,\ \forall t=1,\ldots,T, \tag{8}$$

$$z_{rlt} \leq \overline{Z}_{rl} \quad \forall r=1,\ldots,R,\ \forall l=1,\ldots,L,\ \forall t=1,\ldots,T, \tag{9}$$

$$x_{slt}, x_{ralt}, z_{rlt}, x_{rlt}^{A}, y_{sl\tilde{l}t} \in \mathbb{N} \quad \forall s=1,\ldots,S,\ \forall r=1,\ldots,R,$$
$$\forall a=1,\ldots,A,\ \forall l,\tilde{l}=1,\ldots,L,\ \forall t=1,\ldots,T, \tag{10}$$

$$\mu_{sl}^{A} \in \mathbb{R}_{0}^{+} \quad \forall s=1,\ldots,S,\ \forall l=1,\ldots,L. \tag{11}$$

The objective of the model is to simultaneously maximize the quantity of all analyzed swabs while minimizing the delay, the costs due to the additional mean service rate and the cost to transfer swabs between laboratories, multiplied by the weights δ, α and β, respectively (see the objective function (1)).

Constraint (2) represents the stability condition for the queue; indeed, if, on average, swabs arrivals occur faster than the analysis performed in laboratory (that is, if the swabs to be processed arrive with a greater frequency than those whose analysis has been completed), the queue will grow indefinitely and the system will not have a stationary distribution. Therefore, it is necessary to request that the intensity of traffic is strictly less than 1, $\rho < 1$, that is: $\lambda < \mu$.

Constraint (3) means that the amount of reagent r needed to process all kinds of swabs in the laboratory l at time t is less than or equal to the quantity of reagent r laboratory l receives by each company at time t, summed to the amount of reagent r that laboratory l had placed into storage in the previous period minus what is put in stock at the end of time t, to which we add the quantity of type r reagent that laboratory l is able to self-produce at time t. We underline that, as mentioned above, swab testing often requires not only one type of reagent but a mixture which varies with the type of swab and with the laboratory; hence, constraint (3) must be verified for each needed reagent, the number of s-type swabs that l can analyze depends on the amount of the reagents available to use and, obviously, if in the laboratory l the reagent $\tilde{r}$ is not necessary to process the $\tilde{s}$-type swab, then $p_{\tilde{r}\tilde{s}l} = 0$.

Since the amount of reagent that a laboratory can self-produce at every time is not infinite, constraint (4) establishes the upper bound for each reagent, each laboratory and each time.

Constraints (5)–(6) represent the laboratory capacity that consists in the maximum quantity of swabs (of type s, for constraint (5)) that laboratory l is able to analyze at time t. Such upper bound is defined by the workforce or machinery, including additional ones (which increase the mean service rate). We observe that in constraint (6) the variable is multiplied by a parameter c_{slt} because each swab type can require a different portion of the total laboratory capacity.

Constraint (7) guarantees that no more swabs are processed than required ones (given by the expected value of initial requests to which we add the difference between the swabs to analyze received from other laboratories and that sent to other laboratories).

The inequality (7), in the case in which the swabs are equivalent for the population (as in the case of vaccines), i.e. when the expected value of the quantity of swabs requested from the laboratory l at time t is independent from the swab type s ($\mathbb{E}[q_{lt}]$), can be redefined as follows:

$$\sum_{s=1}^{S} x_{slt} \leq \mathbb{E}[q_{lt}] + \sum_{s=1}^{S} \sum_{\tilde{l}=1}^{L} \left(y_{s\tilde{l}lt} - y_{sl\tilde{l}t} \right) \quad \forall l = 1, \ldots, L, \ \forall t = 1, \ldots, T. \tag{12}$$

Companies (or manufacturing industries and factories) can produce a limited quantity of reagents per period. Therefore, the amount of reagent that company a can send to all laboratories must be less than or equal to that produced, as stated by constraint (8). Furthermore, the sale is often managed by national, regional or local organizations; hence, laboratories do not receive reagents every day (at each period t) but intermittently, as we will assume in Sect. 3, also in order to save on transport costs, considering that the manufacturing companies are often very far from the laboratories.

As mentioned previously, the reagents, or some types of them, require some particular maintenance conditions such as certain temperatures; therefore, constraint (9) establishes that the quantity of reagent of type r that the laboratory l puts into storage

at the end of each period t is less than or equal to the maximum number of stocks of reagent that the laboratory can preserve.

The latest constraints define the domain of the variables of the problem.

3 Numerical Examples

In this section we apply the model to some numerical examples.

Since we want to report all the results for transparency purposes, we select the size of problems as reported and the numerical data are constructed for easy interpretation purposes.

To solve the examples we used Cplex on a laptop with an AMD compute cores 2C+3G processor, 8 GB RAM and we got the best solutions in just a few seconds.

3.1 Base Case with One Period of Time

Firstly, we consider the network consisting of: two companies, two reagents, one laboratory, two swab types and we consider only one period of time.

We assume the following data are given:

$$Q_{111} = 7,\ Q_{121} = 5,\ Q_{211} = 5,\ Q_{221} = 2; \quad p_{111} = 5,\ p_{121} = 1,\ p_{211} = 3,\ p_{221} = 2;$$

$$\mathbb{E}[q_{111}] = \mathbb{E}[q_{211}] = 2; \quad M^A_{111} = M^A_{211} = 0; \quad \overline{M}_{111} = \overline{M}_{211} = 10;$$

$$\overline{M}^A_{11} = \overline{M}^A_{21} = 5; \quad \overline{M}_{11} = 10; \quad c_{111} = 0,8,\ c_{211} = 0.2; \quad \overline{Z}_{11} = \overline{Z}_{21} = 5;$$

$$\delta = \alpha = \beta = 1;\ \mu^i_{11} = \mu^i_{21} = 3; \quad \epsilon_{11} = \epsilon_{21} = 1;\ \vartheta_{111} = \vartheta_{211} = 1;\ z_{110} = z_{210} = 0.$$

The optimal solutions are calculated by solving the optimization problem, the calculations are performed using the Cplex program. We get the following optimal solutions:

$$x_{111} = 1,\ x_{211} = 2; \quad x_{1111} = 7,\ x_{1211} = 0,\ x_{2111} = 5,\ x_{2211} = 2;$$

$$z_{rlt} = x^A_{rlt} = \mu^A_{sl} = y_{sl\tilde{l}t} = 0, \quad \forall s,\ \forall r,\ \forall l,\ \tilde{l},\ \forall t.$$

These optimal solutions clearly show that although 5 units of the first reagent are still available, and although this quantity is enough to process additional swabs, the optimal quantity of swabs that the laboratory can analyze (x_{slt}) is limited by the quantity of the second reagent.

3.2 Case with More Periods of Time

Keeping the same structure of the network and data, now we consider a second example where we suppose two periods of time.

In this case, we also assume:

$$Q_{111} = 10,\ Q_{121} = 8,\ Q_{211} = 8,\ Q_{221} = 2;\ Q_{ra2} = 0,\ \forall r,\ \forall a;$$
$$\mathbb{E}[q_{111}] = 1,\ \mathbb{E}[q_{211}] = 2;\ \mathbb{E}[q_{112}] = 3,\ \mathbb{E}[q_{212}] = 3.$$

In this case (with more than one period of time), it is convenient to stock reagents at the end of time $t = 1$, and use them at time $t = 2$: $z_{111} = 13$, $z_{211} = 12$, $x_{1112} = x_{1212} = x_{2112} = x_{2212} = 0$, also because, as noted in the previous section, companies are often very far from laboratories and the sale is managed by appropriate organizations, therefore, those reagents are not sent every period, but intermittently.

3.3 Case of Epidemic Outbreak and Self-Produced Reagent

As mentioned above, swabs are generally sent at the nearest laboratory (or at the local processing center).

A third example refers to the case when an epidemic outbreak occurs. In this case, the local laboratory receives a number of swabs to be analyzed (usually increasing) which is higher than those that it is able to process. We also introduce the case in which a laboratory can self-produce a reagent.

Now, we consider the network consisting of: one company, two reagents, two laboratories, one swab type and we consider two periods of time.

We indicate below the main data that have undergone a variation compared to the previous examples:

$$Q_{111} = 40,\ Q_{112} = 35,\ Q_{211} = 20,\ Q_{212} = 25;\ p_{111} = 5,\ p_{112} = 0,\ p_{211} = 3,\ p_{212} = 5;$$
$$\mathbb{E}[q_{111}] = 5,\ \mathbb{E}[q_{112}] = 10;\ \mathbb{E}[q_{121}] = 1,\ \mathbb{E}[q_{122}] = 1;$$
$$M^A_{11t} = M^A_{21t} = 0;\ M^A_{12t} = 0,\ M^A_{22t} = 12; \quad \overline{M}^A_{11} = \overline{M}^A_{12} = 10;$$
$$c_{111} = c_{112} = c_{121} = c_{122} = 1; \quad \overline{Z}_{11} = \overline{Z}_{21} = \overline{Z}_{12} = \overline{Z}_{22} = 15;\ \mu^i_{11} = 6,\ \mu^i_{12} = 8.$$

In this case we get the optimal solutions according to which the first laboratory (which is subject to a greater number of requests), sends a portion of swabs to the second laboratory: $y_{1122} = 4$. Furthermore, we observe that in $t = 1$, the second laboratory satisfies the processing requests of the swabs, using only the self-produced reagent: $x_{1121} = 0,\ x_{2121} = 0,\ x^A_{221} = 12$; while in $t = 2$, it uses both the reagent sent by the external manufacturer company and the reagent put into stock

at the previous period (as well as the self-produced reagent): $x_{2122} = 7,\ z_{221} = 7,\ x^A_{222} = 11$.

3.4 Case of Extended Period of Epidemic Outbreak

The fourth example maintains the same structure and data as the previous example, but we add a third time period. We indicate below the added data referring to the period $t = 3$:

$$Q_{113} = 75,\ Q_{213} = 45; \quad \mathbb{E}[q_{113}] = 15,\ \mathbb{E}[q_{123}] = 5.$$

In this case, the optimal solutions clearly show that it is more convenient, for laboratory 1, to use additional mean service rate rather than transferring swabs to the other laboratory: $\mu^A_{11} = 4.1,\ y_{sl\tilde{l}t} = 0,\ \forall s,\ \forall l,\ \tilde{l},\ \forall t$.

Finally, we note that if the cost to raise the service rate, $\epsilon_{sl}(\mu^A_{sl})$, increases from 1 to 10, then, the optimal solution is obtained for:

$$\mu^A_{11} = \mu^A_{12} = 0; \quad y_{1121} = 3,\ y_{1122} = 1,\ y_{1123} = 11.$$

4 Conclusion

COVID-19 is the ongoing disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), a new coronavirus. The virus spreads very quickly and tests for viral presence are used to diagnose positive cases and to allow public health authorities to trace and contain outbreaks. The shortage of reagents and other supplies needed to perform the tests and analyze the swabs, together with the incorrect management of them, has become the main issue for massive testing. Therefore, in this paper we present a multi-period resource allocation model with the main goal of minimizing the impact of a generic pandemic.

The model proposed in this paper allows decision makers to make the best choice, that is, to determine the optimal quantities of: swabs to analyze in each laboratory and at each time, reagents to send by each company to each laboratory at each time, reagents that each laboratory has to put into storage at the end of each period, reagents to self-produce by each laboratory at each time, swabs to transfer between laboratories at each time and the additional mean service rate in order to maximize the number of each type of swab analyzed while minimizing the delay, the costs due to the additional mean service rate and the cost to transfer swabs between laboratories. The experimentation shows that in case of extended periods of time, it is convenient to stock reagents at the end of a time period, and use them subsequently. Furthermore, if a laboratory is located near an area where an outbreak occurs it is affordable to send a portion of swabs to other laboratories;

but if the epidemic outbreak lasts for an extended period of time, then it might be more convenient to use additional mean service rate rather than transferring swabs to other laboratories (depending on the cost to raise the service rate). Finally, we underline that a laboratory could satisfy the processing requests of the swabs, using only the self-produced reagent (if the laboratory is able to do it), but if the quantity of swabs requested to such a laboratory (including those sent from other laboratories) increases significantly, the laboratory could use both the reagent sent by the external manufacturer company and the reagent put into stock at the previous periods (as well as the self-produced reagent).

We created the model in a context of reagents shortage, with the aim of maximizing the number of swabs analyzed in the laboratories, but, by readapting it appropriately, it could be useful for problems in different contexts, for example in the management of mechanical ventilators, vaccines, and so on.

In a future work we intend to continue the study of this topic and, in particular, we aim at analyzing the behavior of the decision makers at all levels of the network, so as to obtain the optimality and equilibrium conditions, and, as a consequence, the global solution for the entire network.

Acknowledgments The research was partially supported by the research project "Programma ricerca di ateneo UNICT 2020–22 linea 2-OMNIA" of Catania. This support is gratefully acknowledged.

References

1. Fernandes, N.: Economic effects of coronavirus outbreak (COVID-19) on the world economy. Available at SSRN 3557504 (2020)
2. Gautam, N., Queueing theory. In: Operations Research and Management Science Handbook, Operations Research Series, 20073432, pp. 1–2 (2007)
3. Keilson, J., Servi, L.D.: A distributional form of Little's Law. Oper. Res. Lett. **7**, 223–227 (1988)
4. Kendall, D.G.: Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded Markov chain. Ann. Math. Stat. 338–354 (1953)
5. Lampariello, L., Sagratella, S.: Effectively managing diagnostic tests to monitor the COVID-19 outbreak in Italy. Technical Report, Optimization Online (2020)
6. Lee, A.M.: A problem of standards of service (Chapter 15). In: Applied Queueing Theory. MacMillan, New York (1966)
7. Little J.D.C.: A proof of the queueing formula $L = \lambda W$. Oper. Res. **9**, 383–387 (1961)
8. Martínez-Álvarez, F., Asencio-Cortés, G., Torres, J.F., Gutiérrez-Avilés, D., Melgar-García, L., Pérez-Chacón, R., Troncoso, A.: Coronavirus optimization algorithm: a bioinspired metaheuristic based on the COVID-19 propagation model. Big Data **8**(4), 308–322 (2020)
9. Mehrotra, S., Rahimian, H., Barah, M., Luo, F., Schantz, K.: A model of supply-chain decisions for resource sharing with an application to ventilator allocation to combat COVID-19. Naval Res. Logist. (NRL) **67**(5), 303–320 (2020)
10. Mizumoto, K., Kagaya, K., Zarebski, A., Chowell, G.: Estimating the asymptomatic proportion of coronavirus disease 2019 (COVID-19) cases on board the Diamond Princess cruise ship, Yokohama, Japan. Eurosurveillance **25**, 2000180 (2020)

11. Prem, K., Liu, Y., Russell, T.W., Kucharski, A.J., Eggo, R.M., Davies, N.: The effect of control strategies to reduce social mixing on outcomes of the COVID-19 epidemic in Wuhan, China: a modelling study. Lancet Public Health **5**, E261–70 (2020)
12. Rawson, T., Brewer, T., Veltcheva, D., Huntingford, C., Bonsall, M.B.: How and when to end the COVID-19 lockdown: an optimization approach. Front. Public Health **8**, 262 (2020)
13. Zonderland, M.E., Boucherie, R.J.: Queuing networks in health care systems. In: Handbook of Healthcare System Scheduling, International Series in Operations Research & Management Science. Springer, Boston (2012)
14. Zhou, Y.-P., Gans, N.: 99-40-B: A Single-Server Queue with Markov Modulated Service Times. Financial Institutions Center, Wharton, UPenn (1999)

Modelling and Solving Patient Admission and Hospital Stay Problems

Rosita Guido, Sara Ceschia, and Domenico Conforti

Abstract This paper considers patient admissions and patient-to-room assignment problems, which are attracting increasing attention. These problems are challenging and concern with the assignment of a set of patients to a set of rooms in a well-defined planning horizon by satisfying several constraints. Hospitals usually face this complex problem of planning admissions and assigning patients to rooms manually, requiring long staff time and high costs. The aim of this paper is to model and solve this problem efficiently. The proposed optimization model is embedded in a matheuristic, which is tested to solve a set of benchmark instances characterized by real-world features. The experimental results show that the solution approach is effective and allows to obtain optimal/sub-optimal solutions in short computational times.

Keywords Combinatorial optimization · Scheduling · Matheuristic · Patient admission

1 Introduction

Hospitals have to face an increasing demand from an ageing population and community health needs. The management of available and appropriate beds and their occupancy levels in hospitals is an integral part of the economical and ethical management of health care [1]. The bottleneck of patient allocation and the availability of beds is one of the major problems within a hospital environment.

R. Guido (✉) · D. Conforti
de-Health Lab, Department of Mechanical, Energy and Management Engineering, University of Calabria, Rende, Italy
e-mail: rosita.guido@unical.it; domenico.conforti@unical.it

S. Ceschia
Polytechnic Department of Engineering and Architecture, University of Udine, Udine, Italy
e-mail: sara.ceschia@uniud.it

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_7

Another important issue is to estimate the patient length of stay (LOS), which is the number of consecutive nights that a patient stays in a department because it affects further admissions. An efficient bed management function is highly dependent on the established patient discharge process. Occupancy levels have to be held high to ensure high utilization of hospital resources. There is thus a critical need to develop more efficient approaches to manage the bed allocation issues affecting hospital staff. This paper focuses on a very challenging task consisting of planning patient admissions and assigning them to hospital rooms.

In the literature, hospital patient admission scheduling problems (PASPs) involve two fundamental decisions: which patient to admit and on which day by taking into account several constraints as bed availability. Zhang et al. [2] highlighted that the first-come-first-served scheduling strategy, which is usually used in hospitals, is not very efficient. Demeester et al. [3] introduced the patient bed assignment problem (PBAP), as a combinatorial optimization problem, that aims at assigning a set of patients to a suitable bed. Patients' admission and discharge dates are known, that is, patients' LOS is well defined. This problem is $\mathcal{NP}$-hard [4] and heuristic approaches, like as simulated annealing [5], column generation [6], matheuristic [7], and late acceptance hill-climbing [8], were developed to solve it. The problem introduced by Ceschia and Schaerf [9] is a dynamic PASP under uncertainty (called PASU), which is more complex than the PBAP. This novel problem introduces for each patient a range of admission and discharge dates and some patients could have fluctuations in lengths of stay because they could extend their LOS. The main decisions concern which patient to admit on a day in the defined range, and how to manage those patients with a risk of overstay. It has hard constraints and soft constraints, which are penalized in the objective function if violated. An optimization model for the online PASU problem was formulated and solved on a set of 450 benchmark instances, with different sizes and complexity, by a simulated annealing approach in [9]. Their results were improved, mainly on the large instances, with an Adaptive Large Neighborhood Search approach in [10]. The above papers considered the online PASU, that is, patients' information is revealed day by day. An optimization model for the offline version of the PASU problem was proposed in [11], where penalty values different from the default ones were tested in order to improve the overall quality of care. In parallel to [9], the PBAP problem was extended in a dynamic context also in [12] modelling and solving it by Integer Linear Programming. Finally, Zhu et al. [13] investigated the problem of designing short-term strategies consistent with the dynamic problem's long-term objective for the PASU problem.

In this paper, we improve the optimization formulation of [11], we propose an efficient matheuristic as a solution approach, and we test it on the small family of the PASU benchmark instances. To evaluate the quality of the solutions, we compute the optimality gap.

2 Problem Statement and a Sparse Optimization Model

In this section, we briefly describe the PASU problem and present a new formulation with a reduced number of variables and constraints that results in a sparse optimization problem to be solved by the matheuristic.

Hospital departments have several levels of expertise (i.e., high, medium, null) in treating a pathology. Rooms are located in departments, they have a fixed number of beds that defines their capacity, they could have some equipment, and be subject to age policies (e.g., geriatric departments) and to gender policies, that is, only patients of a specific gender can be assigned to a given room, or patients with the same gender of patients already staying in the room can be assigned to. The first two policies are called *restricted gender policy* (RGP), whereas the last one is the *dependent gender policy* (DGP). A set of patients is waiting to be admitted to the hospital. For each patient is known the medical pathology, a set of equipment that could be mandatory or preferred, age, gender, and LOS. Some patients could extend their hospital stay. In order to improve patient requirements, transfers of patients among rooms are allowed but not during overstayed periods. The problem has hard and soft constraints, listed in Table 1. The soft constraints can be violated. Their violations are penalized by a related cost coefficient in the objective function.

The PASU problem aims at admitting all patients during their fixed range of admission dates and assigning them to suitable rooms by considering that, however, there are concurrent requests for rooms. The objective function minimizes, as much as possible, soft constraint violations.

2.1 A Sparse Optimization Model for Patient Admission Scheduling Problems Under Uncertainty

In this section, we introduce the used notation and formulate a sparse optimization model for the offline PASU problem. The number of variables and constraints is strongly reduced with respect to the model formulation proposed in [11].

Let P, R, S, E, AP, be the set of patients, rooms, medical specialties, equipment, and age policies, indexed by p, r, sp, e and j, respectively. Let $P^o \subseteq P$, be the set of patients with a risk of overstay; $\bar{S}_r \subseteq S$, be the set of medical specialties that can be treated on room r; and $GP = \{1, 2, 3, 4\}$ be the set of gender policies indexed by gp. The elements 1 and 2 denote RGP; 3 denotes DGP, and 4 that there is no gender policy; $\bar{R}^D = \{r \in R \mid gp_r = 3\}$ is the set of rooms with DGP.

The patient information and the suitably defined sets that allow to reduce the overall number of constraints are reported in Table 2. The decision variables are listed in Table 3.

Table 1 Hard and soft constraints

Constraints	Description
Hard constraints	
H_1—Room capacity	The number of patients assigned to a room cannot exceed the number of beds in the room
H_2—Department specialism	Patients are assigned to rooms where there is a level of expertise in treating the patient's pathology
H_3—Mandatory equipment	A patient has to be assigned to well equipped rooms if he/she has mandatory equipment
H_4—Patient's age	A patient can be assigned to a room only if there is consistency with room-age policy
H_5—Patient admission	Each patient has to be admitted in a certain interval of admission dates
H_6—Hospital stay	For each patient is known the LOS given as the number of consecutive nights
H_7—No transfers in overstay	A patient with overstay risk has to stay in the last assigned room even during an overstay period
Soft constraints	
S_1—Restricted gender policy	The violation concerns male (female) patients assigned to a room for only female (male) patients
S_2—Department specialism	Patients should be assigned to rooms with high level of expertise. Assignments to rooms with a medium level of expertise are penalised
S_3—Room capacity preference	Patients prefer to be assigned to rooms with a specific number of beds (e.g., single, double). The bed capacity of the assigned room should not be greater than the one preferred
S_4—Preferred equipment	Each patient should be assigned to a room with the equipment preferred for the patient
S_5—Dependent gender policy	Male patients should not be assigned to rooms with DGP where there are already female patients, and vice versa
S_6—Transfers	A patient can be transferred from a room to another one for quality of care improvement
S_7—Delay admission	A patient should be admitted on the earliest admission date. Patient admission date can be delayed up to the latest admission date even if penalised
S_8—Overcrowded rooms	A room could be overcrowded only because patients that have an overstay risk cannot be transferred in other rooms

The objective function (1) is the sum of five terms with penalty costs. By referring to the soft constraints described in Table 1, the first term of the objective function penalizes the sum of violations related to Constraints $S_1 - S_4$. The overall violation is penalized by the coefficient w_{pr}. The further four terms of the objective function are

Table 2 Patient's characteristics and notation

$AD_p = \left\{a_p, \ldots, a_p^{'}\right\}$	Range of admission dates where a_p is the earliest date and $a_p^{'}$ is the latest admission date
L_p	Length of hospital stay given as consecutive nights
$DD_p = \left\{z_p, \ldots, z_p^{'}\right\}$	Range of discharge dates, where $z_p = a_p + L_p$ is the earliest discharge date and $z_p^{'} = a_p^{'} + L_p$ is the latest discharge date
$H_p = \left\{a_p, \ldots, z_p^{'} - 1\right\}$	Period during which patient p has a hospital stay
$sp_p \in S$	Patient's specialty
Age	$a_{pj} = 1$ if patient's age is consistent with age policy $j \in AP$, 0 otherwise
$ME_p \subseteq E$	Set of mandatory equipment
$PE_p \subseteq E$	Set of preferred equipment
Gender	Male or female
$no_p \geq 0$	Estimated length of overstay
$H_p^o = \left\{z_p, \ldots, z_p^{'} + no_p - 1\right\}$	Range of discharge dates if p has a risk of overstay, that is, $no_p > 0$
$P_M(P_F)$	Set of male (female) patients
$\bar{R}_p = \{r \in R, j \in AP, e \in E :$	Set of rooms feasible for patient p. It is given as intersection of the constraints on
$sp_p \notin \bar{S}_r, me_{pe} \leq eq_{re}, ap_{rj} a_{pj} > 0\}$	Patient's specialty, mandatory equipment, and age policy
$\bar{R}_p^D = \left\{r \in \bar{R}_p \mid gp_r = 3\right\}$	Subset of rooms with DGP and feasible for patient p

Table 3 Decision variables

$x_{prd} = 1$	If patient p is assigned to room $r \in \bar{R}_p$ on day $d \in H_p$, 0 otherwise
$o_{prd} = 1$	If patient $p \in P^o$ is assigned to room $r \in \bar{R}_p$ on day $d \in DD_p^o$, 0 otherwise
$t_{pd} = 1$	If patient p is transferred on day $d \in H_p$, 0 otherwise
$a_{prd} = 1$	If patient p is admitted on day $d \in Ad_p$ to room $r \in \bar{R}_p$, 0 otherwise
$b_{rd} = 1$	If male and female patients are in room r on day $d \in H_p$, 0 otherwise
$del_p \geq 0$	Delayed admission (in days) for patient p
$ov_{rd} \geq 0$	If room r is overcrowded on day $d \in H$

violations of Constraints S_5, S_6, S_7, S_8 and they are penalised by w_g, w_t, w_{del}, w_o, respectively.

$$\min \sum_{p\in P}\sum_{r\in \bar{R}_p}\sum_{d\in H_p} w_{pr}x_{prd} + \sum_{r\in \bar{R}^D}\sum_{d\in H} w_g b_{rd} + \sum_{p\in P}\sum_{d>a_p}^{d<z_p^{'}} w_t t_{pd} + \sum_{p\in P}(w_{del}del_p)$$
$$+ \sum_{r\in R}\sum_{d\in H} w_o ov_{rd} \tag{1}$$

$$\sum_{d\in AD_p} ad_{pd} = 1 \qquad \forall p \in P \tag{2}$$

$$del_p = \sum_{d\in AD_p} ad_{pd}d - a_p \qquad \forall p \in P \tag{3}$$

$$\sum_{k=d}^{d+L_p-1}\sum_{r\in \bar{R}_p} x_{prk} \geq ad_{pd}L_p \qquad \forall p \in P, d \in AD_p \tag{4}$$

$$\sum_{d\in H_p}\sum_{r\in \bar{R}_p} x_{prk} = L_p \qquad \forall p \in P \tag{5}$$

$$\sum_{r\in \bar{R}_p} x_{prd} \leq 1 \qquad \forall p \in P, d \in H_p \tag{6}$$

$$C_r \geq \sum_{p\in P|d\in H_p, r\in \bar{R}_p} x_{prd} \qquad \forall r \in R, d \in H \tag{7}$$

$$t_{pd} \geq x_{prd} - x_{pr(d-1)} - ad_{pd} \qquad p \in P, r \in \bar{R}_p, d \in AD_p \tag{8}$$

$$t_{pd} \geq x_{prd} - x_{pr(d-1)} \qquad p \in P, r \in \bar{R}_p, d \in]a_p^{'}, z_p^{'}[\tag{9}$$

$$\sum_{d\in]a_p, z_p^{'}[} t_{pd} \leq nt \qquad \forall p \in P \tag{10}$$

$$ad_{pd} + x_{pr(d+L_p-1)} \leq 1 + os_{pr(d+L_p)} \quad \forall p \in P_o, d \in AD_p, r \in \bar{R}_p, d + L_p \leq |H| \tag{11}$$

$$C_r + ov_{rd} \geq \sum_{p\in P|d\in H_p, r\in \bar{R}_p} x_{prd} + \sum_{p\in P_o|d\in DD_p, r\in \bar{R}_p} os_{prd} \quad \forall r \in R, d \in H \tag{12}$$

$$b_{rd} \geq x_{prd} + x_{p'rd} - 1 \quad d \in H, p \in P_{F|d\in H_p}, p^{'} \in P_{M|d\in H_{p'}}, r \in \bar{R}_p^D \cap \bar{R}_{p'}^D \tag{13}$$

Constraints (2) state that each patient has to be admitted in a certain interval of admission dates. Constraints (3) compute a delayed admission with respect to the early date, and Constraints (4)–(6) ensure that the overall stay will occur in a certain number of consecutive L_p nights. Constraints (7) refer to the room capacity constraints (see H_1 in Table 1). Constraints (8)–(9) define if a patient is transferred into another room and Constraints (10) limit the overall number of transfers per patient to nt. Constraints (11) force that a patient will stay in the last room on the overstay. Constraints (12) ensure that a room capacity is not violated during the planning horizon H but a room could be overcrowded only if there are patients who extend their LOS. Finally, Constraints (13) define if the DGP in violated in the rooms $r \in \bar{R}^D$ by setting to one the value of the related b_{rd}.

Hereafter, we refer to Model (1)–(13) together the decision variables listed in Table 3 as Mod^{PASU}.

3 A Matheuristic Approach and Computational Results

The solution approach developed and implemented for solving the PASU problem is based on the matheuristic FiNeMath, which was introduced in [7] for solving the PBAP. FiNeMath showed good performance in terms of results and computational times, as even confirmed if compared to further approaches as the exact one presented in [14]. This matheuristic combines fix and relax heuristic, and fix and optimize heuristic with the large neighborhood search heuristic [15]. Its framework has two fundamental steps: the first step focuses on finding an initial feasible solution of Mod^{PASU}; the second step is devoted to construct a sequence of subproblems, easier to solve than the whole original problem. An overview of FiNeMath is showed in Fig. 1.

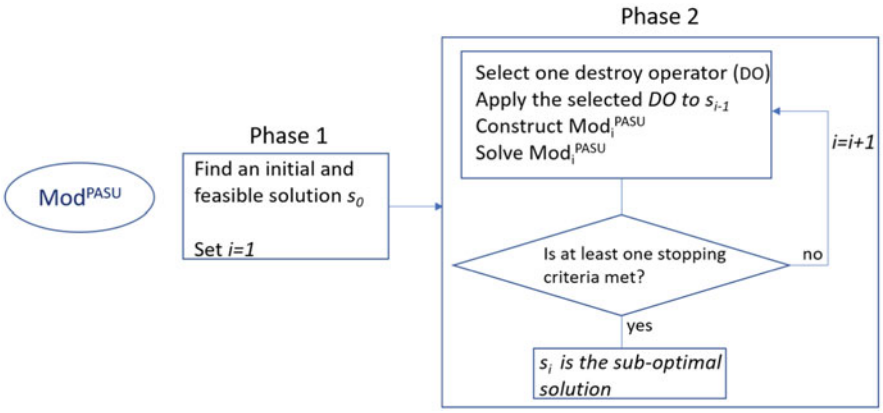


Fig. 1 Overview of the matheuristic approach

This approach is a matheuristic because each subproblem is solved by a MIP solver and is effective even because an incumbent solution is provided as initial solution to start with. At each iteration, a local change is performed by a destroy operator applied to the incumbent solution. The rest of the components of the incumbent solution are added to the Mod^{PASU} as equality constraints.

The new version of FiNeMath here developed is Algorithm 1. At Phase 1, constraints on DGP violation are removed and the patient admission dates are fixed to the earliest in order to find an initial feasible solution in a short computational time. We defined a set $\mathscr{DO}$ of three destroy operators, named DO_1, DO_2, and DO_3. DO_1 is based on removing from the incumbent solution a percentage number of patients randomly selected whereas DO_2 on removing patients with a specific gender; DO_3 combines the above two destroy operators.

Algorithm 1 FiNeMath for PASU problem

Input: Mod^{PASU}, P (set of patients), $\mathscr{DO}$, N_{It}, T_{max}
Output: Phase 1: feasible schedule s_0. Phase 2: final schedule

Phase 1
- Initialisation: $w_g = 0$
- $a_p^{'} \leftarrow a_p \; \forall p \in P$
- Solve the MIP model
- **if** *infeasible* **then**
  - Set $a_p^{'}$ to its original value $\forall p \in P$
  - Solve the Mod^{PASU}

Phase

Phase 2
- Initialisation: $w_g \leftarrow$ its value. $Time \leftarrow 0$
- $a_p^{'} \leftarrow$ its original value $\forall p \in P$
- $i \leftarrow 1$
- **while** $i <= N_{It}$ *or* $Time < T_{max}$ **do**
  - Select one destroy operator in $\mathscr{DO}$ an apply it to the current solution
  - Add the rest of the patient-to-room-day assignements as equality constraints to Mod^{PASU}
  - Solve the new Mod_i^{PASU}
  - $i \leftarrow i + 1$
- **end**

Phase

3.1 Instances and Parameters Setting

We solved the small family of 150 instances,[1] clustered as small-short, small-mid and small-long sets, whose main characteristics are listed in Table 4.

[1] The instances are available at https://bitbucket.org/satt/pasu/.

Table 4 Main characteristics of the Small family of instances and FiNeMath timeout

Set of instances	Departments	Rooms	Equipment	Patients	Specialities	Days	T_{max} (s)
Small short (50)	4	8	4	50	3	14	1000
Small mid (50)	4	8	4	100	3	28	2000
Small long (50)	4	8	4	200	3	56	3000

We set the parameters of Mod^{PASU} to the default values reported in [9]: $w_{pe} = 20$; $w_{cr} = 10$; $w_{sp} = 20$; $w_g = 50$; $w_t = 2$; $w_{del} = 1$; $nt = 1$.

As already observed in [7], the number of assignments added as constraints to Mod^{PASU} influences improvements in the objective function value and computational times: a high percentage of fixed assignments decreases computational times but often reduces objective function improvements. The percentage of destroyed assignments is between 50% and 80%. Each subproblem is solved with a gap of 5%, which decreases up to 1% in the latest iterations. The stopping criteria are the number of carried out iterations $N_{It} = 29$, and the maximum computational time T^{max}, which is family dependent, as reported in Table 4.

3.2 Computational Results and Discussion

In this subsection, we present the computational results related to 150 benchmark instances of the small family. There are three families with 50 instances whose main characteristics are in Table 4. Computational experiments were conducted on a Server running Windows server R2 2012 with Intel Xeon E5-2695v3 14 CORE/64GB. FiNeMath was implemented in ILOG OPL and run with CPLEX 12.7.1, Academic License. The lower bound (LB) values were computed by solving quickly (less than three minutes) Mod^{PASU} as a linear programming model with continuous variables defined in [0, 1].

Per each set of instances, we report the results in terms of mean value in Table 5. The second column shows the objective function value, and the following five columns report the values of the weighted soft constraint violations. The next two columns report the objective function values found in [9, 10], denoted as F^C and F^L, respectively. The following three columns show the LB values, the percentage gap $\frac{F-LB}{F} \times 100$ of F compared to the LB, and the average computational time in seconds. The objective function values are close to the LB values, as a guaranty of high quality of care for patients. The mean percentage gap ΔLB is between 2.32% and 6.12%, and ΔLB is less than 2% for 42 out of the 150 tested instances. Finally, we observe that: (1) the mean optimal value of the small-short set is $\bar{F}^* = 2616.12$ and that the average computational time is 150 s [11]; (2) some of the optimal solutions of the small-mid family were found in 24 h in [9]. This result highlights that FiNeMath is efficient because it reduced computational times, and effective

Table 5 Results as mean values of the instances of the Small family

Family	F	Weighted violations of the soft constraints					F^C	F^L	LB	ΔLB (%)	Time (s)
		$\sum_{i=1}^{4} \bar{v}_i$	$\bar{v}_5$	$\bar{v}_6$	$\bar{v}_7$	$\bar{v}_8$					
Small short	2616.12	2535.20	45.00	4.00	28.00	3.92	2723.02	2694.26	2557.80	2.32	130.12
Small mid	5891.22	5583.60	169.00	6.00	123.28	9.34	6058.18	6119.52	5655.38	5.05	770.08
Small long	11888.90	11128.20	308.00	22.00	408.36	22.36	12260.92	12428.48	11207.86	6.12	1798.98

because it found optimal solutions for all the 50 small short instances and near-optimal solutions for the other 100 instances.

Finally, the F^C and F^L values are greater than F, as expected, because they refer to the online PASU problem which is a dynamic version.

4 Conclusions

In this paper, we formulated a sparse optimization model for offline patient admission planning and scheduling problems and proposed an efficient matheuristic that keeps working in a small feasible region of the large search space. We tested the model and the solution approach on 150 benchmark instances. The results show that the matheuristic minimizes the gap between the found solutions and the corresponding lower bound values found by the exact method.

The number of variables and constraints is strongly reduced with respect to the optimization model formulated in [11]. Indeed, reducing the dimensionality of the problem is critical to improving the performance of the matheuristic. We believe that the proposed optimization model could be used as integrated in a decision support system to help administrators of healthcare institutions to early identify issues in the patients' admission and hospitalization process. In addition, it can be a useful tool for analyzing inefficient management and patient's discomfort. Consequently, alternative solutions, in terms of schedules, could be exploited for organizing hospitalizations that can improve the effectiveness of medical treatments.

Future works are on enhancing the solution approach in order to solve in reasonable computational times large instances and assuring a high quality of solutions.

Acknowledgments The research has been partially supported by the research project "SI.F.I.PA.CRO.DE. Sviluppo e industrializzazione farmaci innovativi per terapia molecolare personalizzata PA.CRO.DE. (PON ARS01_00568, CUP: B29C20000360005, CONCESSIONE RNA-COR: 4646672), Italian Ministry of University and Research, 2021.

References

1. Walsh, M. What is a Bed? Beds as a measure of resource usage and demand. Better Outcomes Health Newsletter. **4**(3) (1998)
2. Zhang, L.M., Chang, H.Y., Xu, R.T.: The patient admission scheduling of an ophthalmic hospital using genetic algorithm. In: Information Technology Applications in Industry, Computer Engineering and Materials Science, vol. 756, pp. 1423–1432 (2013)
3. Demeester, P., Souffriau, W., De Causmaecker, P., Vanden Berghe, G.: A hybrid tabu search algorithm for automatically assigning patients to beds. Artif. Intell. Med. **48**(1), 61–70 (2010)
4. Vancroonenburg, W., Della Croce, F., Goossens, D., Spieksma, F.C.R.: The red-blue transportation problem. Eur. J. Oper. Res. **237**(3), 814–823 (2014)

5. Ceschia, S., Schaerf, A.: Local search and lower bounds for the patient admission scheduling problem. Comput. Oper. Res. **10**(38), 1452–1463 (2011)
6. Troels, M.R., Lusby, R.M., Larsen, J.: A column generation approach for solving the patient admission scheduling problem. Eur. J. Oper. Res. **235**(1), 252–264 (2014)
7. Guido, R., Groccia, M.C., Conforti, D.: An efficient matheuristic for offline patient-to-bed assignment problems. Eur. J. Oper. Res. **268**(2), 486–503 (2018)
8. Bolaji, A.L., Bamigbola, A.F., Shola, P.B.: Late acceptance hill climbing algorithm for solving patient admission scheduling problem. Knowl. Based Syst. **145**, 197–206 (2018)
9. Ceschia, S., Schaerf, A.: Modeling and solving the patient admission scheduling problem under uncertainty. Artif. Intell. Med. **56**(3), 199–205 (2011)
10. Lusby, R.M., Schwierz, M., Range, T.M., Larsen, J.: An adaptive large neighborhood search procedure applied to the dynamic patient admission scheduling problem. Artif. Intell. Med. **74**, 21–31 (2016)
11. Guido, R., Solina, V., Conforti, D.: Offline Patient admission scheduling problems. In: Sforza, A., Sterle, C. (eds.) Optimization and Decision Science: Methodologies and Applications, pp. 129–137. Springer International Publishing, Cham (2017)
12. Vancroonenburg, W., De Causmaecker, P., Vanden Berghe, G.: A study of decision support models for online patient-to-room assignment planning. Ann. Oper. Res. **239**, 253–271 (2016)
13. Zhu, Y.-H., Toffolo, T.A.M., Vancroonenburg, W., Vanden Berghe, G.: Compatibility of short and long term objectives for dynamic patient admission scheduling. Comput. Oper. Res. **104**, 98–112 (2019)
14. Bastos, L., Marchesi, S.L., Janaina, F., Hamacher, S., Fleck, J.L.: A mixed integer programming approach to the patient admission scheduling problem. Eur. J. Oper. Res. **273**(3), 831–840 (2019)
15. Mladenović, N., Hansen, P.: Variable neighborhood search. Comput. Oper. Res. **24**(11), 1097–1100 (1997)

A Two-Stage Variational Inequality for Medical Supply in Emergency Management

Georgia Fargetta and Laura Scrimali

Abstract In this paper, we study the competition of healthcare institutions for medical supplies in emergencies caused by natural disasters. In particular, we develop a two-stage stochastic programming model in a generalized Nash equilibrium framework. It provides the optimal amount of medical supplies from warehouses to hospitals, in order to minimize both the purchasing cost and the transportation costs. For effective disaster planning, we allow for real-time information spreading and up-to-date disaster evaluation. Thus, each institution deals with a two-stage stochastic programming model that takes into account the unmet demand at the first stage, and the consequent penalty. Then, the institutions simultaneously solve their own stochastic optimization problems and reach a stable state governed by the stochastic generalized Nash equilibrium concept. Moreover, we formulate the problem as a two-stage variational inequality. We also present an alternative two-stage variational inequality formulation using the Lagrangian relaxation approximation.

Keywords Disaster management · Stochastic programming · Variational inequality

1 Introduction

In recent years, emergencies and natural disasters have significantly affected our social and economic progress. Therefore, emergency management has become one of the most important and challenging issues. Moreover, emergency resource storage and distribution have led to strong competition for medical supplies among healthcare institutions.

G. Fargetta · L. Scrimali (✉)
Department of Mathematics and Computer Science, University of Catania, Catania, Italy
e-mail: georgia.fargetta@phd.unict.it; laura.scrimali@unict.it

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_8

In this paper, we investigate hospital competition for medical supplies as a generalized Nash equilibrium problem, and propose a stochastic programming model to describe the behaviour of each demand location. Thus, we are able to obtain the optimal amount of medical items from warehouses to hospitals, in order to minimize both the purchasing cost and the transportation costs. Following [20], we consider real-time information spreading and up-to-date disaster evaluation. Therefore, we provide a two-stage stochastic programming model based on disaster scenarios that takes into account the unmet demand at the first stage, and the consequent penalty, see [11]. In particular, in the first stage, hospitals receive the early warning information about the emergency and decide the medical item procurement planning; however, they are not aware of the real situation. Subsequently, accurate real-time information is observed and the process reaches the second-stage, where the decision relies on the first-stage solution and on the observed scenario. Moreover, we introduce a penalty function for the unmet demand of medical supplies in the second stage decision, see also [6]. The problem is then formulated as a two-stage stochastic variational inequality (see [10]).

The importance of an efficient approach to emergency management and medical supply planning has been investigated in several papers. For instance, in [15] the authors construct a generalized Nash equilibrium model with stochastic demand to analyse competition among organizations for medical supplies. The problem is then formulated as a variational inequality, using the concept of variational equilibrium. In [14], Nagurney et al. present a stochastic generalized Nash equilibrium model for disaster relief. Each humanitarian organization solves a two-stage stochastic optimization problem, and the model is formulated as a finite-dimensional variational inequality. In [19], Nagurney and Salarpour introduce a variational inequality formulation of a two-stage stochastic game theory model in order to examine the behavior of national governments during Covid-19 pandemic and their competition for essential medical supplies. In [11], the authors develop a stochastic programming model to select the storage locations of medical supplies and require inventory levels for each type of medical supply. The resulting model captures the information updating making use of disaster scenarios. In [3], the authors present an optimization model consisting of a dynamic supply chain network for personal protective equipment, and study the related evolutionary variational inequality in the presence of a delay function.

Recently, two-stage stochastic variational inequalities have been introduced, where one seeks a decision vector before the stochastic variables are known, and a decision vector after the scenario has been realized. In [2], the authors propose a two-stage stochastic variational inequality model to deal with random variables in variational inequalities, and formulate this model as a two-stage stochastic programming with recourse. In [10], the authors investigate the transformation of a general two-stage stochastic programming problem to a two-stage stochastic variational inequality. In [16], Rockafellar and Wets discuss the multistage stochastic variational inequality. In [17], the authors develop progressive hedging methods for solving multistage convex stochastic programming, see also [18].

In this paper, we extend the model in [15] in two directions. First, we tackle the medical item procurement planning problem as a two-stage stochastic programming problem. Then, we describe the model as a generalized Nash equilibrium problem. Our second improvement is the characterization as a two-stage variational inequality. This approach allows us to decompose the problem in two lower dimensional variational inequalities, instead of solving a unique large-scale variational inequality. We also present an alternative formulation based on Lagrangian relaxation approximation, that makes it possible to investigate the role of Lagrange multipliers in the market behavior.

The structure of this paper is as follows. In Sect. 2, we introduce the two-stage stochastic model. In Sect. 3, we model the competition among healthcare institutions as a generalized Nash equilibrium problem, and provide a two-stage variational inequality formulation. We also present an alternative two-stage variational inequality based on the Lagrangian relaxation approach. Finally, in Sect. 4, we draw our conclusions and present further research issues.

2 Two-Stage Stochastic Model of the Competition for Medical Supply

In this section, we present a two-stage stochastic model for the medical supply competition. Let $\mathcal{W}$ denote the set of warehouses, with typical warehouse denoted by w; let $\mathcal{H}$ denote the set of hospitals, with typical hospital denoted by h; let $\mathcal{K}$ denote the set of medical supply type, with typical type denoted by k, and let $\mathcal{M}$ denote the set of transportation modes, with typical mode denoted by m. We consider a network representation as in Fig. 1. The links between the levels of the network represent all the possible connections between warehouses and hospitals. Multiple links between each warehouse and each hospital depict the possibility of alternative modes of transportation. We note that the choice of the transportation

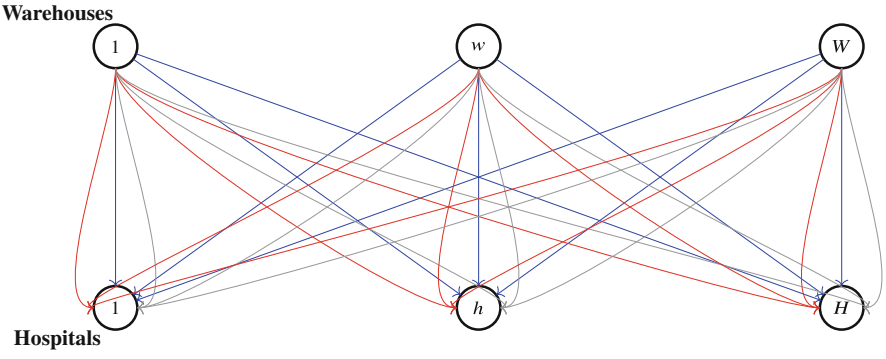


Fig. 1 The network representation

mode is due to the distance between supply and demand locations. For example, for long distances airplanes are preferred to transportation by truck or train. The choice of the transportation mode may also depend on the type of medical item or on the severity of the emergency.

We denote by Q_w^k the amount of medical item of type k in warehouse w, and by $Q_w = \sum_{k\in\mathcal{K}} Q_w^k$ the total amount of medical items in warehouse w.

Now, let x_{wh}^k be the amount of medical item of type k from warehouse w to hospital h, and let ρ_w^k be the unitary price of medical item k at warehouse w. Let x_{wh} denote the total amount delivered from warehouse w to hospital h, where

$$x_{wh} = \sum_{k\in\mathcal{K}} x_{wh}^k.$$

We further group the x_{wh} into the WH-dimensional column vector x.

In addition, we introduce the transportation time t_{wh}^m from warehouse w to hospital h with mode m and assume that it depends on the amount x_{wh}, namely, $t_{wh}^m = t_{wh}^m(x_{wh})$.

Table 1 summarizes the relevant notations used in the model formulation.

We consider a pre-event policy, in which each demand location (hospital) seeks to minimize the purchasing cost of medical items and the transportation time from the first stage, and a recourse decision process to optimize the transportation costs from the second stage, in response to each disaster scenario. Let $(\Omega, \mathcal{F}, P)$ be a probability space, where the random parameter $\omega \in \Omega$ represents the typical disaster scenario. For each $\omega \in \Omega$, we denote by $\xi : \Omega \to \mathbb{R}^{WHK+HK}$ a finite dimensional random vector and by $\mathbb{E}_\xi$ the mathematical expectation with respect to ξ. In order to formulate the two-stage stochastic model, we introduce two types of decision variables. The first-stage decision variable x_{wh}^k is used to represent the quantity of medical supplies of type k from warehouse w to hospital h. The second-stage decision variables are $y_{wh}^k(\omega)$ and $z_h^k(\omega)$. The variable $y_{wh}^k(\omega)$ represents the quantity of medical supplies of type k to be delivered from warehouse w to hospital h under scenario ω. The variable $z_h^k(\omega)$ is the unfulfilled demand at hospital h of medical item k under scenario ω. We penalize the amount of unfulfilled demand $z_h^k(\omega)$ by function $\pi_h^k = \pi_h^k(z_h^k(\omega), \omega)$. From the perspective of demand locations, x_{wh}^k is chosen before a realization of ξ is revealed and later $y_{wh}^k(\omega)$ and $z_h^k(\omega)$ are selected with known realization. Finally, we introduce the transportation cost c_{wh}^m from warehouse w to hospital h with mode m and assume that it depends on the amount $y_{wh}(\omega) = \sum_{k\in\mathcal{K}} y_{wh}^k(\omega)$, namely, $c_{wh}^m = c_{wh}^m(y_{wh}(\omega), \omega)$.

Our aim is to obtain an efficient plan of medical item procurement of each demand location in the first stage by the evaluation of adaptive plans in the second stage.

Table 1 The notation for the two-stage stochastic model

Symbols	Definitions
$\mathcal{W}$	Set of warehouses, with typical warehouse denoted by w, $card(\mathcal{W}) = W$
$\mathcal{H}$	Set of hospitals, with typical hospital denoted by h, $card(\mathcal{H}) = H$
$\mathcal{K}$	Set of different medical items, with typical item denoted by k, $card(\mathcal{K}) = K$
$\mathcal{M}$	Set of transportation modes, with typical mode denoted by m, $card(\mathcal{M}) = M$
d_h^k	Demand of medical item k of hospital h in stage one
$d_h^k(\omega)$	Demand of medical item k of hospital h in stage two under scenario ω
x_{wh}^k	Amount of medical item k from warehouse w to hospital h in stage one
x_{wh}	Amount of medical items delivered from warehouse w to hospital h in stage one
x	Amount of total medical items from all warehouses to all hospitals in stage one
Q_w^k	The amount of the medical item k in warehouse w
$Q_w = \sum_{k\in\mathcal{K}} Q_w^k$	The total amount of the medical items in warehouse w
e_k	Maximum amount available of medical item k
ρ_w^k	Unitary price of medical item k at warehouse w
$t_{wh}^m(x_{wh})$	Transportation time from warehouse w to hospital h with mode m
$y_{wh}(\omega)$	Amount of medical items to be delivered from warehouse w to hospital h in stage two under scenario ω
$z_h^k(\omega)$	Amount of unfulfilled demand at hospital h of medical supply item k under scenario ω
$c_{wh}^m(y_{wh}(\omega), \omega)$	Transportation cost from warehouse w to hospital h with mode m under scenario ω
$\pi_h^k(z_h^k(\omega), \omega)$	Penalty for unfulfilled demand at hospital h of medical supply item k under scenario ω

2.1 First-Stage Problem

For each hospital h, we minimize the purchasing cost and the transportation time of the first stage with the expected overall costs and the penalty for the prior plan. Therefore, the first-stage problem is given by:

$$\min \sum_{w\in\mathcal{W}} \left(\sum_{k\in\mathcal{K}} \rho_w^k x_{wh}^k + \sum_{m\in\mathcal{M}} t_{wh}^m(x_{wh}) \right) + \mathbb{E}_\xi(\Phi_h(x, \xi(\omega))) \tag{1}$$

subject to

$$\sum_{h\in\mathcal{H}} \sum_{k\in\mathcal{K}} x_{wh}^k \le Q_w, \ \forall w \in \mathcal{W}, \tag{2}$$

$$\sum_{w \in \mathcal{W}} x_{wh}^k = d_h^k, \ \forall k \in \mathcal{K}, \tag{3}$$

$$\sum_{w \in \mathcal{W}} x_{wh}^k \leq e_k, \ \forall k \in \mathcal{K}, \tag{4}$$

$$x_{wh}^k \geq 0, \ \forall w \in \mathcal{W}, \ \forall k \in \mathcal{K}. \tag{5}$$

The objective function (1) minimizes the sum of the purchasing cost for early supply plan, the transportation time, and the expected value of costs of hospital h in the second stage with respect to disaster scenarios. Constraint (2) ensures that the storage capacity of warehouse w is satisfied. It is a shared constraint and realizes that hospitals compete for medical items available at the warehouses at a maximum supply. Constraint (3) states that the amount of delivered medical items of type k has to satisfy the requirement of hospital h; constraint (4) is the maximum availability constraint of medical item k; constraint (5) is the non-negativity requirement on variables. We require that $d_h^k \leq e_k, \ \forall h, k$. Finally, we assume that $t_{wh}^m(\cdot)$ is continuously differentiable and convex for all w, h, m.

2.2 Second-Stage Problem

The second stage is the evaluation of the first stage to obtain the optimal medical supply procurement. In this stage, real-time disaster information is revealed, and hospital h adapts his plan by taking recourse decisions.

For a given realization $\omega \in \Omega$, the second-stage problem of hospital h is given as:

$$\Phi_h(x, \xi(\omega)) = \min \sum_{w \in \mathcal{W}} \sum_{m \in \mathcal{M}} c_{wh}^m(y_{wh}(\omega), \omega) + \sum_{k \in \mathcal{K}} \pi_h^k(z_h^k(\omega), \omega) \tag{6}$$

subject to

$$\sum_{h \in \mathcal{H}} \sum_{k \in K} y_{wh}^k(\omega) \leq Q_w(\omega) - \sum_{h \in \mathcal{H}} \sum_{k \in \mathcal{K}} x_{wh}^k, \ \forall w \in \mathcal{W}, P\text{-}a.s., \tag{7}$$

$$\sum_{w \in \mathcal{W}} y_{wh}^k(\omega) + z_h^k(\omega) = d_h^k(\omega), \ \forall k \in \mathcal{K}, P\text{-}a.s., \tag{8}$$

$$\sum_{w \in \mathcal{W}} y_{wh}^k(\omega) \leq e_k(\omega), \ \forall k \in \mathcal{K}, P\text{-}a.s., \tag{9}$$

$$y_{wh}^k(\omega) \geq 0, z_h^k(\omega) \geq 0 \ \forall w \in \mathcal{W}, \ \forall k \in \mathcal{K}, P\text{-}a.s. \tag{10}$$

Thus, $\Phi_h(x, \xi(\omega))$ is the optimal value of the second-stage problem (6)–(10), where the constraints hold almost surely (P-a.s.). The objective function (6) minimizes the total cost and the penalty for the unmet demand at the second stage. Constraint (7) is the warehouse storage capacity which takes into account the quantity of medical items already delivered. Constraint (8) is a balance constraint, and states that the supply at the second stage plus the unmet demand should be equal to the demand at the second stage. Constraint (9) is the maximum availability constraint of medical supply of type k at the second stage. Finally, (10) is the non-negativity constraint. We emphasize that the connection between stage-wise decision variables x and y is captured by coupling constraint (7). It is the linking factor between the first and second stage, and communicates the first-stage decisions to the second one.

We assume that:

(a) $c^m_{wh}(\cdot, \omega)$, $\pi^k_h(\cdot, \omega)$, a.e. in Ω, are continuously differentiable and convex for all w, h, k, m;
(b) for each $u \in \mathbb{R}^{WH}$, $c^m_{wh}(u, \cdot)$ is measurable with respect to the random parameter in Ω for all w, h, m;
(c) for each $v \in \mathbb{R}^{HK}$, $\pi^k_h(v, \cdot)$ is measurable with respect to the random parameter in Ω for all h, k;
(d) $y^k_{wh} : \Omega \to \mathbb{R}$ and $z^k_h : \Omega \to \mathbb{R}$ are measurable mappings for all w, h, k;
(e) $d^k_h : \Omega \to \mathbb{R}$ is a measurable mapping for all h and all k.

Finally, we require that $d^k_h(\omega) \leq e_k(\omega)$, $\forall h, k$ and for all scenario ω.

If the random parameter $\omega \in \Omega$ follows a discrete distribution with finite support $\Omega = \{\omega_1, \ldots, \omega_r\}$ and probabilities $p(\omega_r)$ associated with each realization ω_r, $r \in \mathcal{R} = \{1, \ldots, R\}$, then the two-stage problem of hospital h can be formulated as the unique large scale problem:

$$\min \sum_{w \in \mathcal{W}} \left(\sum_{k \in \mathcal{K}} \rho^k_{wh} x^k_{wh} + \sum_{m \in \mathcal{M}} t^m_{wh}(x_{wh}) \right)$$

$$+ \sum_{r \in \mathcal{R}} p(\omega_r) \left(\sum_{w \in \mathcal{W}} \sum_{m \in \mathcal{M}} c^m_{wh}(y_{wh}(\omega_r), \omega_r) + \sum_{k \in \mathcal{K}} \pi^k_h(z^k_h(\omega_r), \omega_r) \right) \tag{11}$$

subject to

$$\sum_{h \in \mathcal{H}} \sum_{k \in \mathcal{K}} x^k_{wh} \leq Q_w, \ \forall w \in \mathcal{W}, \tag{12}$$

$$\sum_{w \in \mathcal{W}} x^k_{wh} = d^k_h, \ \forall k \in \mathcal{K}, \tag{13}$$

$$\sum_{w \in \mathcal{W}} x^k_{wh} \leq e_k, \ \forall k \in \mathcal{K}, \tag{14}$$

$$\sum_{h\in\mathcal{H}}\sum_{k\in\mathcal{K}} y_{wh}^k(\omega_r) \leq Q_w(\omega_r) - \sum_{h\in\mathcal{H}}\sum_{k\in\mathcal{K}} x_{wh}^k,\ \forall w\in\mathcal{W}, \forall r\in\mathcal{R}, \tag{15}$$

$$\sum_{w\in\mathcal{W}} y_{wh}^k(\omega_r) + z_h^k(\omega_r) = d_h^k(\omega_r),\ \forall k\in\mathcal{K}, \forall r\in\mathcal{R}, \tag{16}$$

$$\sum_{w\in\mathcal{W}} y_{wh}^k(\omega_r) \leq e_k(\omega_r),\ \forall k\in\mathcal{K}, \forall r\in\mathcal{R}, \tag{17}$$

$$x_{wh}^k \geq 0,\ \forall w\in\mathcal{W},\ \forall k\in\mathcal{K}, \tag{18}$$

$$y_{wh}^k(\omega_r) \geq 0,\ \forall w\in\mathcal{W},\ \forall k\in\mathcal{K}, \forall r\in\mathcal{R}, \tag{19}$$

$$z_h^k(\omega_r) \geq 0,\ \forall k\in\mathcal{K}, \forall r\in\mathcal{R}. \tag{20}$$

3 Stochastic Generalized Nash Equilibrium

Competition for medical supplies among hospitals can be studied also as a game. The underlying equilibrium concept is then that of a stochastic generalized Nash equilibrium (SGNE), namely, a Nash equilibrium when the functions are expected value functions, and the players are subject to shared constraints.

We define the sets:

$$S_h = \left\{x_h = (x_{wh}^k)_{w,k} \in \mathbb{R}^{WK} : (3) - (5) \text{ hold}\right\},$$

$$X = \{x = (x_h)_h \in \mathbb{R}^H : x \text{ satisfies } (2)\},$$

$$T_h = \left\{(y_h(\omega), z_h(\omega)) = \left((y_{wh}^k(\omega))_{w,k}, (z_h^k(\omega))_k\right) \in \mathbb{R}^{WK+K} : (8) - (10) \text{ hold, } P\text{-}a.s.\right\},$$

$$V = \{(y(\omega), z(\omega)) = (y_h(\omega), z_h(\omega))_h \in \mathbb{R}^{2H} : (7) \text{ holds, } P\text{-}a.s.\}.$$

We also define $S = \prod_h S_h$ and $T = \prod_h T_h$.

We refer to the objective function (1) for $h \in \mathcal{H}$ as the function:

$$\mathbb{J}_h(x_h, x_{-h}) = \sum_{w\in\mathcal{W}} \left(\sum_{k\in\mathcal{K}} \rho_{wh}^k x_{wh}^k + \sum_{m\in\mathcal{M}} t_{wh}^m(x_{wh}) \right) + \mathbb{E}_\xi(\Phi_h(x_h, x_{-h}, \xi(\omega))),$$

where x_{-h} denotes the amount of medical items required by all hospitals except for h.

Definition 1 A vector of medical items $x^* = (x_h^*, x_{-h}^*) \in S \cap X$ is a stochastic generalized Nash equilibrium of the first-stage if for each $h \in \mathcal{H}$

$$\mathbb{J}_h(x_h^*, x_{-h}^*) \leq \mathbb{J}_h(x_h, x_{-h}^*), \quad \forall x_h \in S_h, \forall x \in X.$$

Analogously, we can define the SGNE for the second stage. A solution to such a problem can be found solving a quasi-variational inequality; see [7]. We point out that this problem can also be solved in the form of a variational inequality, using the concept of variational equilibrium; see [5, 9]. However, we note that it is not possible to obtain a full characterization of the solutions of a SGNE problem as solutions of a variational inequality. For this reason, recently, some authors focused on the computation of non-variational equilibria; see [4, 12].

3.1 Two-Stage Variational Inequality Formulation

We now adopt a variational equilibrium approach to our SGNE problem. The two-stage stochastic problem is then equivalent to a two-stage variational inequality (see [8, 13] for theoretical aspects and applications on variational inequality theory).

We restrict our attention to the case of discrete probability distribution.

Theorem 1 *The vector* $(x^*, y^*(\omega_r), z^*(\omega_r))$, $\forall \omega_r, r \in \mathcal{R}$, *is an optimal solution of the medical item procurement planning if and only if:*

1. *the vector* $x^* = (x_h^*, x_{-h}^*) \in S \cap X$ *is a solution of the variational inequality*

$$\sum_{w\in\mathcal{W}}\sum_{h\in\mathcal{H}}\sum_{k\in\mathcal{K}}\left(\rho_{wh}^k + \sum_{m\in\mathcal{M}}\frac{\partial t_{wh}^m(x_{wh}^*)}{\partial x_{wh}^k} + \sum_{r\in\mathcal{R}} p(\omega_r)\frac{\partial \Phi_h(x^*, \xi(\omega_r))}{\partial x_{wh}^k}\right) \times (x_{wh}^k - x_{wh}^{*k}) \geq 0,$$

$$\forall x \in S \cap X; \tag{21}$$

2. *the vector* $(y^*(\omega_r), z^*(\omega_r)) \in T \cap V$, $\forall \omega_r, r \in \mathcal{R}$, *is a solution of the variational inequality*

$$\sum_{r\in\mathcal{R}} p(\omega_r) \sum_{w\in\mathcal{W}}\sum_{h\in\mathcal{H}}\sum_{k\in\mathcal{K}}\left(\sum_{m\in\mathcal{M}}\frac{\partial c_{wh}^m(y_{wh}^*(\omega_r), \omega_r)}{\partial y_{wh}^k(\omega_r)}\right) \times (y_{wh}^k(\omega_r) - y_{wh}^{*k}(\omega_r))$$

$$+ \sum_{r\in\mathcal{R}} p(\omega_r) \sum_{h\in\mathcal{H}}\sum_{k\in\mathcal{K}}\frac{\partial \pi_h^k(z_h^{*k}(\omega_r), \omega_r)}{\partial z_h^k(\omega_r)} \times (z_h^k(\omega_r) - z_h^{*k}(\omega_r)) \geq 0,$$

$$\forall (y(\omega_r), z(\omega_r)) \in V \cap T. \tag{22}$$

In order to ensure the existence of solutions to (21), we note that the set $S \cap X$ is compact and convex, and the operator that enters (21) is continuous. Thus, a solution exists from the standard theory of variational inequalities; see [8]. A similar reasoning ensures the existence of solutions to (22).

3.2 Lagrangian Relaxation Approach

In the following, we provide an alternative two-stage variational inequality, that leads to a lower bound to the optimal value of the initial model. We relax constraints (2) and (7) into their respective objective functions by Lagrangian relaxation approach (see [1, 10]). We associate a non-negative Lagrange multiplier $\lambda_w \geq 0$ to constraint (2), for each $w \in \mathcal{W}$. We group all the Lagrange multipliers into the vector $\lambda \in \mathbb{R}^W_+$. Analogously, we associate a non-negative Lagrange multiplier $\mu_w(\omega_r) \geq 0$ to constraint (7), for each $w \in \mathcal{W}$ and $r \in \mathcal{R}$. We group all the Lagrange multipliers into the vector $\mu(\omega_r) \in \mathbb{R}^W_+$, $\forall r \in \mathcal{R}$. Thus, we find the following two-stage variational inequality:

1. Find $x^* = (x^*_h, x^*_{-h}) \in S$ and $\lambda^* \in \mathbb{R}^W_+$ such that

$$\sum_{w\in\mathcal{W}}\sum_{h\in\mathcal{H}}\sum_{k\in\mathcal{K}}\left(\rho^k_{wh} + \sum_{m\in\mathcal{M}}\frac{\partial t^m_{wh}(x^*_{wh})}{\partial x^k_{wh}} + \sum_{r\in\mathcal{R}} p(\omega_r)\frac{\partial \Phi_h(x^*, \xi(\omega_r))}{\partial x^k_{wh}} + \lambda_w\right) \times (x^k_{wh} - x^{*k}_{wh})$$
$$+ \sum_{w\in\mathcal{W}}\left(Q_w - \sum_{h\in\mathcal{H}}\sum_{k\in\mathcal{K}} x^{*k}_{wh}\right) \times (\lambda_w - \lambda^*_w) \geq 0, \forall x \in S, \lambda \in \mathbb{R}^W_+. \tag{23}$$

2. Find $(y^*(\omega_r), z^*(\omega_r)) \in T$ and $\mu^*(\omega_r) \in \mathbb{R}^W_+$, $\forall \omega_r, r \in \mathcal{R}$, such that

$$\sum_{r\in\mathcal{R}} p(\omega_r)\sum_{w\in\mathcal{W}}\sum_{h\in\mathcal{H}}\sum_{k\in\mathcal{K}}\sum_{m\in\mathcal{M}}\left(\frac{\partial c^m_{wh}(y^*_{wh}(\omega_r), \omega_r)}{\partial y^k_{wh}(\omega_r)} + \mu_w(\omega_r)\right) \times (y^k_{wh}(\omega_r) - y^{*k}_{wh}(\omega_r))$$
$$+ \sum_{r\in\mathcal{R}} p(\omega_r)\sum_{h\in\mathcal{H}}\sum_{k\in\mathcal{K}}\frac{\partial \pi^k_h(z^{*k}_h(\omega_r), \omega_r)}{\partial z^k_h(\omega_r)} \times (z^k_h(\omega_r) - z^{*k}_h(\omega_r))$$
$$+ \sum_{r\in\mathcal{R}} p(\omega_r)\sum_{w\in\mathcal{W}}\left(Q_w(\omega_r) - \sum_{h\in\mathcal{H}}\sum_{k\in\mathcal{K}}(x^k_{wh} + y^{*k}_{wh}(\omega_r))\right) \times (\mu_w(\omega_r) - \mu^*_w(\omega_r)) \geq 0$$
$$\forall (y(\omega_r), z(\omega_r)) \in T, \forall \mu(\omega_r) \in \mathbb{R}^W_+. \tag{24}$$

The operator in (23) can be obtained applying Karush-Kuhn-Tucker conditions to the Lagrangian relaxation of the problem (1)–(5), with dual variable λ_w, for all $w \in \mathcal{W}$. Thus, the first term is the stationarity condition of each optimization problem (1)–(5); while the second term is the complementarity condition. Analogously, we can construct the operator in (24).

We emphasize that variational inequalities (23)–(24) are expressed in terms of Lagrange variables λ_w and $\mu_w(\omega_r)$, that have a fundamental role in regulating the medical item procurement. In fact, λ_w is a control variable on the item availability level; whereas $\mu_w(\omega_r)$ is a control variable on the second-stage warehouse storage capacity. Therefore, the above formulation can be advantageous since allows us to gain a deeper understanding of the market behavior.

4 Conclusions

In this paper, we present a stochastic generalized Nash equilibrium model for a medical supply network. Specifically, we consider a two-layer network that consists of warehouses and hospitals with multiple medical items and multiple transportation modes. Each hospital solves a two-stage stochastic optimization problem: in the first stage, he seeks to minimize the purchasing cost of medical items and the transportation time; in the second stage, he adopts a recourse decision process to optimize the expected overall costs and the penalty, for the prior plan for each possible disaster scenario. The hospitals simultaneously solve their own stochastic optimization problems and reach a stable state given by the stochastic generalized Nash equilibrium concept. The model is formulated as a two-stage variational inequality. By Lagrangian relaxation approximation, an alternative two-stage variational inequality based on the Lagrange multipliers is given. We highlight that dual variables play a fundamental role in investigating the market behavior.

Further research issues are the study of the two-stage stochastic problem in the case of general probability distribution with the associated infinite-dimensional variational inequality, and a characterization of the second-stage equilibrium by means of infinite-dimensional Lagrange duality tools. Partial results in these directions have already been achieved.

Acknowledgments The research was partially supported by the research projects "Problemi di equilibrio: metodi variazionali e teoria dei giochi" GNAMPA-INdAM and "Programma ricerca di ateneo UNICT 2020–22 linea 2-OMNIA" University of Catania. These supports are gratefully acknowledged.

References

1. Beraldi, P., Bruni, M.E., Conforti, D.: A solution approach for two-stage stochastic nonlinear mixed integer programs. Algorithm. Oper. Res. **4**, 76–85 (2009)
2. Chen, X., Pong, T.K., Wets, R.J.-B.: Two-stage stochastic variational inequalities: an ERM-solution procedure. Math. Program. **165**, 1–41 (2017)
3. Daniele, P., Sciacca, D.: A dynamic supply chain network for PPE during the Covid-19 pandemic. J. Appl. Numer. Optim. **3**, 403–424 (2021)
4. Facchinei, F., Sagratella, S.: On the computation of all solutions of jointly convex generalized Nash equilibrium problems. Optim. Lett. **5**(3), 531–547 (2011)
5. Facchinei, F., Fischer, A., Piccialli, V.: On generalized Nash games and variational inequalities. Oper. Res. Lett. **35**, 159–164 (2007)
6. Fargetta, G., Scrimali, L.: Optimal emergency evacuation with uncertainty. In: Parasidis, I.N., Providas, E., Rassias, Th.M. (eds.) Mathematical Analysis in Interdisciplinary Research. Springer Optimizationa and its applications, Vol. 179, Springer, (2021). (to appear)
7. Harker, P.T.: Generalized Nash games and quasi-variational inequalities. Eur. J. Oper. Res. **54**, 81–94 (1991)
8. Kinderlehrer, D., Stampacchia, G.: An Introduction to Variational Inequalities and Their Applications. Academic Press, New York (1980)

9. Kulkarni, A.A., Shanbhag, U.V.: On the variational equilibrium as a refinement of the generalized nash equilibrium. Automatica **48**, 45–55 (2012)
10. Li, M., Zhang, C.: Two-Stage Stochastic variational inequality arising from stochastic programming. J. Optim. Theory Appl. **186**, 324–343 (2020)
11. Mete, H.O., Zabinsky, Z.B.: Stochastic optimization of medical supply location and distribution in disaster management. Int. J. Prod. Econ. **126**(1), 76–84 (2010)
12. Nabetani, K., Tseng, P. Fukushima, M.: Parametrized variational inequality approaches to generalized Nash equilibrium problems with shared constraints. M. Comput. Optim. Appl. **48**(3), 423–452 (2011)
13. Nagurney, A.: Network Economics: A Variational Inequality Approach (2nd ed. (revised)). Kluwer Academic Publishers, Boston (1999)
14. Nagurney, A., Salarpour, M., Dong, J., Nagurney, L.S.: A stochastic disaster relief game theory network model. SN Oper. Res. Forum **1**(10), 1–33 (2020)
15. Nagurney, A., Salarpour, M., Dong, J., Dutta, P.: Competition for medical supplies under stochastic demand in the Covid-19 pandemic: a generalized Nash equilibrium framework. In: Rassias, Th.M., Pardalos, P.M. (eds.) Nonlinear Analysis and Global Optimization. Springer Optimization and Its Applications, pp. 331–356. Springer, Cham (2021)
16. Rockafellar, R.T., Wets, R.J.-B.: Stochastic variational inequalities: single-stage to multistage. Math. Program. **165**, 1–30 (2016)
17. Rockafellar, R.T., Sun, J.: Solving monotone stochastic variational inequalities and complementarity problems by progressive hedging. Math. Program. **174**, 453–471 (2019)
18. Rockafellar, R.T., Sun, J.: Solving Lagrangian variational inequalities with applications to stochastic programming. Math. Program. **181**, 435–451 (2020)
19. Salarpour, M., Nagurney, A.: A multicountry, multicommodity stochastic game theory network model of competition for medical supplies inspired by the covid-19 pandemic. Int. J. Prod. Econ. **236**, 108074 (2021)
20. Wang, L.: A two-stage stochastic programming framework for evacuation planning in disaster responses. Comput. Ind. Eng. **145**, 106458 (2020)

Part III
Scheduling and Planning

The Value of the Stochastic Solution in a Two-Stage Assembly-to-Order Problem

Paolo Brandimarte, Edoardo Fadda, and Alberto Gennaro

Abstract We consider a simple assembly to order problem, where components must be manufactured under demand uncertainty and end items are assembled only after demand is realized. The problem can be naturally cast as a two-stage stochastic linear program with recourse, and possibly generalized to multiple stages. We investigate the two-stage case not only because it make sense, e.g., in specific newsvendor-like applications, but also because it allows a thorough investigation of relevant issues. In this paper we investigate the conditions under which using a stochastic programming approach yields significant advantages over a straightforward deterministic model based on the expected value of demand. The analysis is carried out on the basis of a large number of out-of-sample scenarios, assessing the so-called value of the stochastic solution. We study the impact of problem features such as demand variability, skewness and multimodality, number of specific components, profit margin, and capacity tightness. Furthermore, we compare the behavior of standard two-stage stochastic programming against linear decision rules.

Keywords Stochastic programming · Production planning · Linear decision rules

1 Introduction and Motivation

In production management, risk due to demand uncertainty can be hedged by safety stocks or capacity buffers. Another possibility is risk pooling by common components. Here we consider flat bills of materials, comprising only two levels:

P. Brandimarte (✉)
Dipartimento di Scienze Matematiche, Politecnico di Torino, Torino, Italy
e-mail: paolo.brandimarte@polito.it

E. Fadda · A. Gennaro
Politecnico di Torino, Torino, Italy
e-mail: edoardo.fadda@polito.it; alberto.gennaro@studenti.polito.it

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_9

end items and components (modules). Even though demand for end items may be highly variable, demand for common components (i.e., components that are used in multiple end items) may be less variable, an idea that is exploited in Assembly-To-Order (ATO) manufacturing environments. An ATO strategy is sensible when the long lead time to manufacture or procure components makes a pure Make-To-Order approach not feasible, and a pure Make-To-Stock approach is ruled out by the large number of end item configurations that arise by combining even a relatively small number of components. If final assembly is relatively fast, we may produce components under demand uncertainty, while delaying final assembly until customer orders are actually received. The strategy is common in the automotive industry and is exploited by some computer manufacturers; see [10] for an application to the HP DeskJet printer case.

1.1 Paper Positioning

There is a considerable amount of literature dealing with variations of classical stochastic models for inventory management. In simple cases, one can obtain analytical results and the idea has been applied to systems with assembly operations. An alternative approach to planning within an ATO setting relies on stochastic programming with recourse, two-stage or multistage. Indeed, the modeling flexibility and the freedom in scenario generation allows to overcome some limitations of classical stochastic inventory models.

An early reference based on stochastic programming is [9], which applies scenario aggregation to small problem instances within a two-stage integer programming framework, where the second stage is component allocation. They consider up to 30 scenarios and no more than 5 end items and components. More recent papers include, e.g., [5] and [6]. In the first paper, a multiperiod inventory control problem is tackled within a two-stage framework. The first-stage problem aims at optimizing the parameters of a base-stock policy. They deal with an inventory management problem subject to deterministic lead times, rather than a capacitated manufacturing problem. The second paper is similar in vein and uses specific properties of chained bill of materials, where component requirements can only take values 0 or 1, to carry out a mathematically elegant analysis. The structure of the Bill Of Materials (BOM) differs in the two papers. In one case it is a W structure, where three types of components, two common and one specific, are used to produce two end items. In the other case it is an M structure, where two types of components are used to produce three end items; one end item needs both components, whereas the other two end items need one type of component. Given this simple structure, exact solution procedures may be devised for Poisson demand. We stress the fact that, although their problem involves multiple periods, the approach is two-stage, and aims at finding control parameters for continuous review policies under compound Poisson demand. In [12], emphasis is again on structural properties for problems

with including a small number of items (two). In [4], emphasis is on solving ATO planning problems involving integer variables.

The approach pursued in the present paper is completely different. We consider production planning subject to capacity constraints, for cases including 200 items and 150 components. Also the number of scenarios is larger, up to 300. In order to do so with a limited computational effort, we consider two-stage continuous LP models. Two-stage models are relevant in some settings akin to newsvendor models. In fashion, a quite limited number of decision stages is involved, resulting in newsvendor-like models [7], which may also be used to decompose multistage problems [11, Chapter 5]. We do not deal with integer programming problems, assuming that the involved volumes justify a continuous LP approach, which is standard, e.g., in the lot-sizing literature. In view of a possible extension to truly multistage cases (not just two-stage multiperiod ones) we also investigate the application of decision rules in the vein of [8].

We do not strive for mathematical elegance or structural results, not because they are not important, but because we have other priorities. The most important point, indeed, is that we should not take for granted that the use of sophisticated stochastic optimization models is warranted. To the contrary, [3] shows that a model based on expected values of demand may perform remarkably well. This depends not only on the problem structure, but also on the specific problem features. We should investigate which factors contribute most to the value of using a stochastic programming model, like demand variability, skewness, and multimodality, the number of specific versus common components, profit margin, and manufacturing capacity tightness. All of these issues are disregarded in the aforementioned papers, which have a different aim. To accomplish the task we need to pursue an out-of-sample analysis, in order to estimate the Value of the Stochastic Solution (VSS) [1] under different problem settings. Such an out-of-sample analysis is quite demanding in a multistage setting.

Plan of the Paper In Sect. 2 we describe possible decision models for the ATO planning problem. In Sect. 3 we outline a few computational experiments to give a feeling for the impact of problem features and the relative performance of alternative decision models. Finally, in Sect. 4, we draw preliminary conclusions and describe ongoing and future research.

2 The Decision Models

In this section we describe four different decision models:

- A two-stage stochastic LP model with recourse, which we will refer to as Recourse Problem (RP).
- A simplified deterministic LP model where we ignore demand uncertainty and use expected values of demand, which we will refer to as Expected Value problem (EV).

- A straightforward model based on Linear Decision Rules, referred to as LDR problem.
- A refinement of the LDR model, based on Deflected Linear Decision Rules, referred to as DLDR problem.

In order to state the decision models describing the ATO production planning problem, let us introduce the following sets: the set of components $\mathcal{I} = \{1, \ldots, I\}$, the set of end items $\mathcal{J} = \{1, \ldots, J\}$, the set of production resources (machine groups) $\mathcal{M} = \{1, \ldots, M\}$, and the set $\mathcal{S} = \{1, \ldots, S\}$ of scenarios that we use to discretize the distribution of random demand. We denote by d_j^s the demand for end item $j \in \mathcal{J}$ in scenario $s \in \mathcal{S}$; the probability of scenario s is denoted by π^s. In the following, we will consider Monte Carlo scenario sampling; hence, $\pi^s = 1/S$ and scenarios are equiprobable, but this need not be the case.

Furthermore, let us introduce the following parameters:

- C_i, cost of component $i \in \mathcal{I}$, and P_j, price of the end item $j \in \mathcal{J}$;
- L_m availability (in terms of time) of machine group $m \in \mathcal{M}$;
- T_{im} processing time for component $i \in \mathcal{I}$ on machine $m \in \mathcal{M}$;
- G_{ij} number of components of type $i \in \mathcal{I}$ needed to assemble one end item of type $j \in \mathcal{J}$; in manufacturing parlance, these numbers are called *gozinto factors*. Since we deal with flat bills of materials, they are fully specified by the matrix of gozinto factors.

Note that the limited capacity of machine groups refers only to the production of components. In an ATO environment, final assembly should not be a bottleneck, so we disregard resource constraints at the assembly level.

Finally, we need to introduce two sets of decision variables:

- the first-stage variables x_i, the amount of component i that we produce;
- the second-stage variables y_j^s, the amount of end item j assembled and sold in scenario s, after observing actual demand.

We will experiment with a continuous linear program and not an integer one; hence, the above variables are just required to be non-negative. We do so in order to ease the computational burden, without affecting the conclusions significantly. This is a common simplification, for instance, in the lot-sizing literature and it may be justified when demand volume is large enough, since rounding effects are negligible.

Then, the resulting RP model is:

$$\max \quad -\sum_{i \in \mathcal{I}} C_i x_i + \sum_{s \in \mathcal{S}} \pi^s \left(\sum_{j \in \mathcal{J}} P_j y_j^s \right) \tag{1}$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{I}} T_{im} x_i \leq L_m \qquad \forall m \in \mathcal{M} \tag{2}$$

$$y_j^s \leq d_j^s \qquad \forall j \in \mathcal{J},\ s \in \mathcal{S} \tag{3}$$

$$\sum_{j \in \mathcal{J}} G_{ij} y_j^s \leq x_i \qquad \forall i \in \mathcal{I},\ s \in \mathcal{S} \tag{4}$$

$$y_j^s, x_i \geq 0 \qquad \forall j \in \mathcal{J},\ s \in \mathcal{S},\ i \in \mathcal{I}. \tag{5}$$

The objective function of the problem is the expected net profit, expressed in (1) as expected revenue at the second stage minus cost at the first stage. Constraints (2) limit the machine availability; constraints (3) state that it is not possible to sell more than demand, and constraints (4) preclude assembling items for which we lack the necessary components, thereby linking the two decision stages. Constraints (5) are the usual non-negativity conditions.

Since dealing with uncertainty potentially implies a considerable increase in computational effort, it is important to compare the obtained performance against the much simpler model in which uncertainty is ignored and it is assumed that the expected demand $\bar{d}_j$ will be realized. In this case, we assume that what is assembled is also sold, and the assembly decision variable is just y_j. The resulting EV model is:

$$\max \quad -\sum_{i \in \mathcal{I}} C_i x_i + \sum_{j \in \mathcal{J}} P_j y_j \tag{6}$$

$$\text{s.t.} \quad y_j \leq \bar{d}_j \qquad \forall j \in \mathcal{J} \tag{7}$$

$$\sum_{j \in \mathcal{J}} G_{ij} y_j \leq x_i \qquad \forall i \in \mathcal{I}, \tag{8}$$

also subject to (2) and (5).

In principle, second-stage variables should be functions mapping random outcomes to decisions. When dealing with continuous random variables, this means that second-stage variables are actually infinite-dimensional objects. Discrete scenario generation is one way to make the idea computationally viable, but it becomes critical in a multistage setting due to the explosion of the scenario tree. Hence, a possible alternative is to assume a simple functional form in mapping outcomes to decisions. The simplest mapping is linear, which leads to a linear (affine) decision rule approach [8], where we keep the first stage decisions x_i and map deviations of demand with respect to the expected values into assembly decisions. The linear affine mapping is specified by the following parameters:

- $\bar{y}_j$ is the number of end item j produced under a nominal condition, where demand assumes its expected value;
- $H_{j,k}$ is the rate at which the number of assembled end items j changes as a function of a deviation in demand for item k.

Then, the LDR model is obtained by replacing the scenario-dependent second-stage variable y_j^s by

$$\bar{y}_j + \sum_{k \in \mathcal{J}} H_{jk}\left(d_k^s - \bar{d}_k\right), \tag{9}$$

which is constrained to be within the range $[0, d_j^s]$. The form of this affine rule looks more complicated than in other applications, due to the interaction of different end items through components. Actually, we should introduce a coefficient H_{jk} only when end items j and k share a common component. Note that there is a potentially large number of such coefficients, and that the model structure may be less amenable to efficient interior point methods, resulting in a non-negligible CPU time for its solution.

A further issue with an LDR approach is its rigidity, which may partially eased by adopting deflected (or piece-wise) linear decision rules, which rely on asymmetric deviations $d_j^{+s} = \max(d_j^s - \bar{d}_j, 0)$ and $d_j^{-s} = \max(\bar{d}_j - d_j^s, 0)$, $s \in \mathcal{S}$, $j \in \mathcal{J}$. Then, we replace the second-stage decisions y_j^s by

$$\bar{y}_j + \sum_{k \in \mathcal{J}} \left(H_{jk}^{+} d_k^{+s} + H_{jk}^{-} d_k^{-s} \right). \tag{10}$$

The resulting DLDR decision model suffers from the same issues as the LDR model. Nevertheless, it is worth investigating these approaches, given the potential advantage in a multistage setting.

3 Numerical Experiments

In this section we outline some numerical results obtained by the different decision models. Since space limitations preclude the description of a full-fledged experimental design, we prefer to give some glimpse in order to get a feeling for the problem and the impact of its features.

We generate bills of materials (BOMs) randomly, setting the number of families and common and specific components. In Fig. 1 we illustrate the structure of BOMs by a two-dimensional heat map, where colors are associated to the number of components for each end item. We consider 150 end items and 200 components. End items are listed on rows, and the matrix looks block-diagonal, where blocks correspond to families. We clearly see the presence of common components in each block. The bar chart below the heat map gives the number of produced components (from the solution of the RP model), and there is a spike corresponding to common components.

It is important to emphasize that, in the presence of several common components, a simple model based on expected values of demand may perform fairly well.

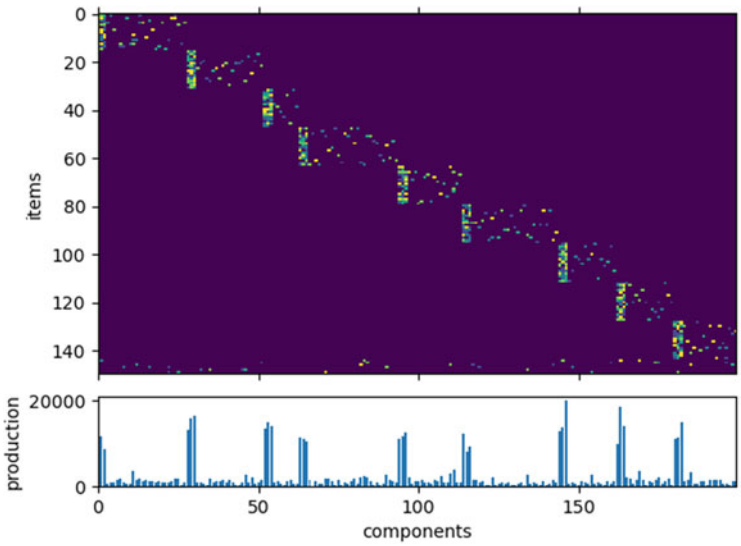


Fig. 1 Structure of BOMs

This also depends on the tightness of capacity constraints and the profit margins. In order to get a clue, in Table 1 we compare the out-of-sample performance of first-stage plans obtained by simpler models (expected value and deflected decision rules) against the full-fledged recourse model. We solve the recourse model with a suitable number of scenarios (more on this later), and then assess its out-of-sample performance on 20,000 scenarios, where we solve the second stage problem for a known demand and a given first-stage plan for component production. For each problem setting, the optimization is run over 100 random instances. The number reported is the percentage loss with respect to the RP model, measured by $(\overline{O}_{RP} - \overline{O}_{xx})/\overline{O}_{RP}$ where $\overline{O}_{RP}$ is the average out-of-sample profit from the RP model, and $\overline{O}_{xx}$ is the average out-of-sample profit from EV or DLDR. Hence, when the number is larger than 100%, it means that the simpler model obtains a negative profit on average.

Problem settings differ in terms of profit margin, capacity tightness, and number of specific versus common components. In the first column, LM and MM refer to the profit margin, which is either low or medium. More precisely, we consider the average return from an end item, which is the range $[0.05, 0.2]$ for low margins and $[0.2, 0.4]$ for medium margins. Low and medium margins are more troublesome than high margins, in the sense that not accounting for uncertainty in these settings has a larger impact. Capacity tightness is set to 0.8 and 1.3; this factor is used to generate available capacity for each of five machine groups on the basis of expected required capacity, which is just obtained by translating expected end item demand to the corresponding machining time on each machine group. Demands for end items are assumed independent and identically distributed, which is clearly not

Table 1 Percentage loss with respect to recourse formulation under different distribution and conditions (profit margins, tightness, number of common and specific components) for expected value (EV) and deflected linear decision rules (DR) models

Parameters				Normal		Uniform		beta(1;4)		beta(4;1)		Normal_mix	
PM	t	cc	sc	EV	DR	EV	DR	EV	DR	EV	DR	EV	DR
LM	0.8	0	10	73.57	13.43	77.02	9.8	140.45	8.1	19.14	3.63	1346.08	70.83
LM	0.8	0	30	63.75	10.17	91.25	10.48	136.12	8.66	20.45	5.02	1588.48	75.54
LM	0.8	3	10	50.05	4.67	51.89	4.92	106.44	7.61	15.13	1.21	1235.94	36.0
LM	0.8	3	30	43.95	5.28	44.0	4.38	94.34	5.14	19.94	1.73	1306.2	34.87
LM	1.3	0	10	94.76	13.72	98.38	9.06	203.54	9.77	22.69	6.07	1726.46	70.98
LM	1.3	0	30	87.97	12.89	106.88	8.55	179.62	9.48	19.25	5.44	2159.75	57.53
LM	1.3	3	10	79.74	4.64	81.76	4.99	135.76	6.54	20.82	2.02	1916.17	33.11
LM	1.3	3	30	68.74	4.51	83.19	4.53	146.98	6.32	23.1	1.81	1538.11	37.74
MM	0.8	0	10	35.86	10.68	44.83	7.56	96.28	8.7	12.45	3.79	756.87	66.72
MM	0.8	0	30	42.63	10.66	47.63	7.84	93.09	9.49	10.82	3.18	998.66	56.33
MM	0.8	3	10	33.88	2.65	33.07	3.54	67.77	3.8	12.66	1.29	765.77	32.55
MM	0.8	3	30	35.64	3.0	34.22	3.62	56.66	4.52	11.7	1.57	674.72	38.22
MM	1.3	0	10	47.22	9.38	53.4	8.59	105.83	10.51	12.41	3.31	974.46	69.09
MM	1.3	0	30	49.82	10.14	47.59	6.27	107.29	10.23	10.0	3.12	951.84	75.19
MM	1.3	3	10	42.67	2.99	47.46	3.65	81.48	5.66	14.28	1.24	1058.37	30.77
MM	1.3	3	30	42.82	4.15	51.47	3.53	82.45	5.08	13.43	0.93	1086.77	31.53

realistic, but allows us to better see the impact of the distribution. The distribution can be symmetric (truncated normal or uniform), left-skewed (a shifted and scaled beta distribution with parameters 1 and 4), right-skewed (a shifted and scaled beta distribution with parameters 4 and 1), and a bimodal mixture of normals. In our case we mix two normals $\mathsf{N}(100, 50^2)$ and $\mathsf{N}(1000, 300^2)$, with 50% probability each. Thus, the expected value of demand is 550 and, given the standard deviations 50 and 300, there is a limited overlap of tails. The intuition provided by the newsvendor problem [2], where the optimal order quantity is given by a quantile of the demand distribution, and the quantile depends on the problem economics, suggests that replying on the expected demand (550) is going to be a disaster, unless the economics are favorable.

It is important to realize that the results heavily depend on the problem features and on how BOMs are generated, but some messages are clear from the table.

- There may be a significant loss by disregarding uncertainty, and there is a price to be paid for rigidity of deflected linear rules. We do not report results for the plain LDR model, which are poorer.
- The distribution has a huge impact and, in some problem settings, the performance of EV is more than disappointing. For instance, in the beta (1,4) case, there is a skew to the right, which implies that the EV model is optimistic and produces too many components that are then scrapped. Component scrapping is huge in the case of the bimodal, as expected demand is in the no man's land between the two normals, and the impact on profit is severe as the economics are not favorable.
- This effect is more relevant at high capacity levels, where we are free to produce more components, and hurts most in the case of low margins. In the limit, with a very small capacity inducing a lot of lost sales, demand uncertainty would be irrelevant, and a simple newsvendor model suggests that with high margin we may afford scrapping unused components.

It is also worth noting that some disappointing results that we observe may occur because of component scrapping, which also raises environmental concerns beyond plain and simple expected profit. This is relevant in a two-stage setting; less so in a multistage extension.

In order to get a feeling for the involved risk and the out-of-sample performance of first-stage plans, we depict a histogram of out-of-sample profits in Fig. 2. In the uniformly distributed case displayed on the left, it is quite clear that the EV model may even result in negative profit, and that DLDR is outperformed by RP. We do not show the EV model performance in the bimodal case displayed on the right, as profit is quite negative. Here we use a hybrid of RP and DLDR, as 50% of end items are dealt with by scenario dependent second stage variables and 50% are dealt with by DLDR, but this is not quite relevant.

Another obvious question is how many scenarios are needed to capture demand uncertainty adequately. To get a clue, we may solve the RP model for an increasing the number of in-sample scenarios, and then plot the corresponding out-of-sample average profit. Figure 3 shows that indeed average profit increases by adding

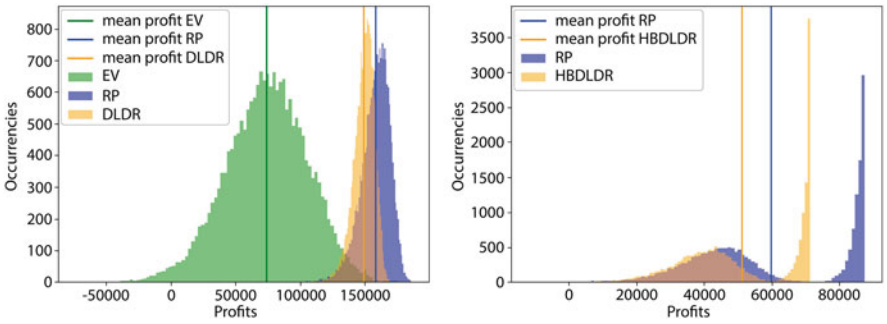


Fig. 2 Comparing out-of-sample profits for different decision models for a uniform (picture on the left) and a multimodal (picture on the right) distribution

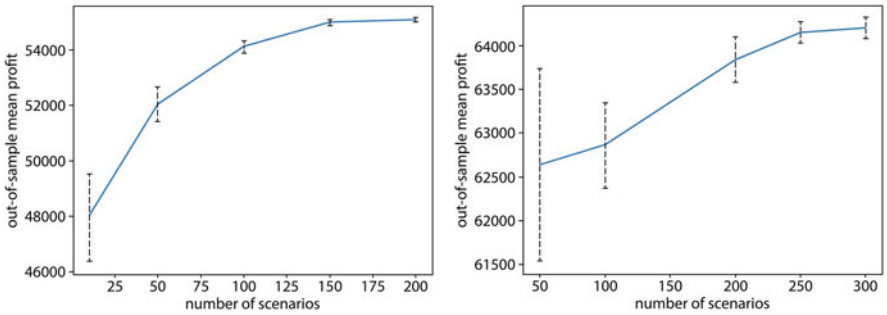


Fig. 3 Impact of number of scenarios in the case of a uniform (picture on the left) and a multimodal (picture on the right) distribution

scenarios, but there is a saturation effect with a limited number of scenarios. Again, the precise behavior depends on the specific problem features, but this seems to suggest that we do not necessarily need a huge number of scenarios to obtain satisfactory results. Rather unsurprisingly, we need more scenarios in the case of a multimodal distribution (displayed on the right) than in the case of a uniform distribution (displayed on the left). The CPU time to solve these problem instances is a few seconds, which may sound reassuring. However, this is not likely to carry over to the multistage case, or when we introduce risk measures.

Given the perspective of an extension to multistage problems, we may also wonder if we could reduce the number of scenarios by disregarding some end items for which uncertainty is not quite relevant. For this subset of items, we could just use the expected value of demand and sample demand for the remaining items. To define the subset of items for which we consider demand uncertainty, we may rank end items according to a given metric, and represent demand uncertainty only for the top subset. We may consider two simple metrics, one (called margin) based on end item profitability, and another one (called sc) based on a suitable measure of component specificity, where the rationale is that items with more specific components are

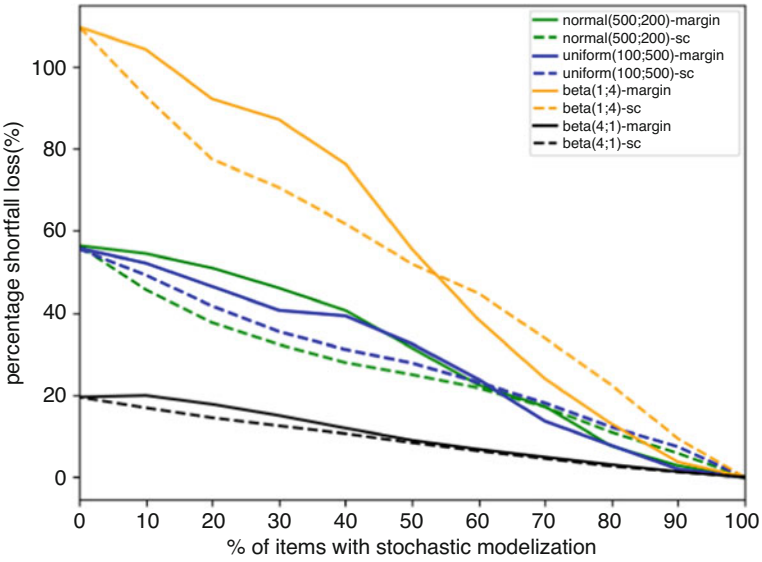


Fig. 4 Loss, with respect to full representation of uncertainty, of margin and sc scenario generation policies

more critical. In Fig. 4 we plot the percentage loss of these two scenario generation policies, for a varying percentage of stochastic end items, with respect to the profit of a fully stochastic model. When the percentage is 0%, we essentially have the EV model, whereas we have the RP model when the percentage is 100%. This experiment shows again the huge impact of specific demand distributions. Unfortunately, there is no evident "knee" in the plots, showing that accounting for uncertainty is relevant (or that the simple policies are just ineffective). In the two stage case, the issue is not quite relevant as a moderate number of scenarios is effective, but the issue is open for the multistage case.

4 Conclusions and Current/Future Research

We have discussed simple models for two-stage production planning in an ATO environment. Clearly, the results from synthetic instances must be taken with great care, but it is clear that under some problem settings the cost of ignoring uncertainty can be huge. Problem features, like demand variability, skew, and multimodality, profit margin, and capacity tightness have a significant impact. Piecewise linear decision rules do not seem to perform very well for this problem, but a rather surprising fact is that a limited number of scenarios is enough to considerably improve the results with respect to the EV model. Clearly, the ATO problem is different from a financial optimization problem, as common components hedge

against uncertainty and capacity limitations moderate the advantage of perfect information. This finding should be tempered however.

- We did not consider correlations: we could have negative correlations inside a product family (where demand substitution is to be expected) and positive correlation between families (possibly because of market penetration of the firm or general economic conditions).
- More scenarios would be needed if we want to introduce risk measures, like conditional value-at-risk, which focus on bad tails.
- Scenario generation remains quite critical for multistage problems. Since linear decision rules do not perform well in the two-stage case, maybe alternative strategies should be considered, possibly based on approximate dynamic programming and time-based decomposition.
- The two stage problem makes sense in a fashion setting, as we mentioned, where not only risk measure should be considered, but also distributional ambiguity, which we disregarded here.

All of these issues are the subject of our current research.

References

1. Birge, J., Louveaux, F.: Introduction to Stochastic Programming, 2nd edn. Springer, New York (2011)
2. Brandimarte, P., Zotteri, G.: Introduction to Distribution Logistics. Wiley, Hoboken (2007)
3. Caro, F., Gallien, J.: Clearance pricing optimization for a fast-fashion retailer. Oper. Res. **60**, 1404–1022 (2012)
4. DeValve, L., Pekeč, S., Wei, Y.: A primal-dual approach to analyzing ATO systems. Manag. Sci. **66**, 5389–5407 (2010)
5. Doğru, M.K., Reiman, M.I., Wang, Q.: A stochastic programming based inventory policy for assemble-to-order systems with application to the W model. Oper. Res. **58**, 849–864 (2010)
6. Doğru, M.K., Reiman, M.I., Wang, Q.: Assemble-to-order inventory management via stochastic programming: Chained BOMs and the M-system. Prod. Oper. Manag. **26**, 446–468 (2017)
7. Fisher, M., Raman, A.: Reducing the cost of demand uncertainty through accurate response to early sales. Oper. Res. **44**, 87–99 (1996)
8. Georghiou, A., Kuhn, D., Wiesemann, W.: The decision rule approach to optimization under uncertainty: methodology and applications. Comput. Manag. Sci. **16**, 545–576 (2019)
9. Jönsson, H., Jörnsten, K., Silver, E.A.: Application of the scenario aggregation approach to a two-stage, stochastic, common component, inventory problem with a budget constraint. Eur. J. Oper. Res. **68**, 196–211 (1993)
10. Lee, H.L., Billington, C.: Material management in decentralized supply chains. Oper. Res. **41**(5), 835–847 (1993)
11. Nahmias, S.: Production and Operations Management, 4th edn. McGraw–Hill, New York (2001)
12. Wang, X.J.: Stochastic programming formulations and structural properties for assemble-to-order systems. Ph.D. thesis, McMaster University (2020). http://hdl.handle.net/11375/25464

Robust Optimal Planning of Waste Sorting Operations

Diego Maria Pinto, Claudio Gentile, and Giuseppe Stecca

Abstract Waste management and circular economy objectives are worthwhile and worldwide challenges concerning both the protection of the environment and the conservation of natural resources with aim of zero waste. A considerable attention has been directed over the last decade towards the optimization of planning procedures related to waste management in order to empower circular economy ambition. This work investigates the operations of waste recycling centers where materials are collected by a fleet of trucks and then sorted in order to be converted in secondary raw materials. The activity is characterized by low margins, difficulties to track flows and uncertainties in supplies. In a previous work by Pinto and Stecca a formulation has been proposed to address and optimize the sorting process. However, special attention should be paid to the fact that waste streams processes are affected by several uncertainties, such as the stochastic processes regarding waste arrivals to sorting facilities. This work extends the above mentioned formulation by introducing robustness to data uncertainties related to waste supplies. Accordingly, the main aim of this study is to develop a mixed integer linear programming model for planning and scheduling the packaging waste recycling operations taking into consideration also the stochastic nature of waste arrivals. This is done by introducing a protection function in each constraint according to the probabilistic robust approach presented in (Bertsimas and Sim, Oper Res 52(1):35–53, 2004). This approach ensures deterministic and probabilistic guarantees on constraints satisfaction. The model supports other strategic decisions such as sizing of the amount of processed waste and allocation of the optimal number of operators for each shift of the waste sorting processes. Experiments are performed on instances taken from a real case scenario in Italy and comparisons are made against different planning strategies.

D. M. Pinto (✉) · C. Gentile · G. Stecca
Istituto di Analisi dei Sistemi ed Informatica “A. Ruberti”, Consiglio Nazionale delle Ricerche, Rome, Italy
e-mail: diegomaria.pinto@iasi.cnr.it; gentile@iasi.cnr.it; giuseppe.stecca@iasi.cnr.it

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_10

Keywords Robust optimization · Circular economy · Lot sizing · Mixed integer linear programming

1 Introduction

Performances of waste management systems have been improving thanks to a noticeable commitment of decision makers and research efforts regarding the optimization of each system component. As an example, a conventional operational task addressed by research is about waste shipment and collection trucks route optimization as in [3, 6]. In the meantime, similar optimization models have been drastically reducing transportation costs enhancing the growth of the online shopping of any sort of good. As a result, while logistic companies start serving a new magnitude of customers, also a new dimension of packaging waste started affecting the overall waste system. This leads to the need of a stronger technological and strategic decision support to packaging waste facilities in order to lower all the extra costs involved with the selective collection and sorting of this kind of waste. Not only logistic companies but also every other kind of industry generates a considerable amount of packaging waste. In Europe the Directive 2004/12/EC on packaging and packaging waste laid down the European recycling and recovery targets. In particular, official reporting on packaging waste for all EU Member States was implemented in 2007 and since then Eurostat monitors also the developments of this important statistics. Therefore, the need of meeting the recovery and recycling targets imposed by EU law and the rising prices of raw materials used for packaging have resulted in an increasing interest in the recovery of materials from the waste streams. Moreover, the recycling industry is characterized by very low margins and high percentage of operation and logistics costs. For this reason it is critical the optimization of the process in order to turn it into an economically sustainable business. Special attention should be paid to the fact that this objective is affected by several uncertainties such as those arising in the waste streams processes. In particular, waste arrivals to sorting facilities are stochastic processes. Indeed, waste truck arrivals are subject to considerable variability that should be properly addressed when modeling scenarios including waste streams. In [5] this subject has been investigated. This work intends to expand the modeling power of the MILP presented in [5] by introducing robustness to data uncertainties related to waste supplies. Accordingly, the main research aim of this study is to develop a mixed integer linear programming model for planning and scheduling the packaging waste recycling operations taking into consideration also the stochastic nature of waste arrivals. This is done by introducing a protection function in some constraints according to the probabilistic robust approach presented in [2]. This approach ensures deterministic and probabilistic guarantees on constraints satisfaction and it does so in a straightforward framework. The model supports also other strategic decisions such as sizing the amount of processed waste and allocating the optimal number of operators for each shift of the waste sorting processes.

The reminder of the paper is organized as follows: the literature review is given in Sects. 1.1; Sect. 2 is dedicated to the problem description and the MILP formulation; Sect. 3 presents the experimental results; Sect. 4 gives conclusions and future research perspectives.

1.1 Literature Review

The range of scientific literature contributions to waste management is justified by the variety of technological configurations and decision levels (mainly strategic and operational). The main type of waste flows considered are municipal solid waste, as a result the great majority of works are related to the management or the strategic definition of municipal solid waste networks, such as in [7]. The conventional operational task addressed by research is about collection trucks route optimization and waste shipment [3, 6]. Besides them, a real case application is presented in [1]. A complete survey of both strategic and tactical issues in solid waste management that have been addressed by operations research methods is presented in [4]. Within the waste management paradigm, none of the previous works is close to the original operational application of mathematical programming presented in [5]. This paper intends to expand the value of the original work by extending the presented model with the introduction of robustness to data uncertainties related to waste supplies.

2 Problem Definition and Modeling

In this section the main operational features covered by the nominal deterministic model are described together with the formulation of its robust counterpart. We will clarify how the model is able to cover the main strategic decisions of the process while properly modeling the typical production dynamics of a reverse logistic setting.

The production demand of the waste facility arises from the need to program and size the sorting operations of waste in order to balance the availability of the buffer of received material with the production and set-up costs of sorting operations and storage costs of all the inter-operational buffers. Therefore, the simultaneity of the scheduling problem and the lot sizing problem is highlighted.

It is important to notice that, in the considered industrial case, costs of storage are not measurable directly. Indeed, it is impossible to compute the inventory costs as proportional to the inventory value because there is no means to evaluate that value before the material is sorted. At the same time the level of buffer storage can be such as to constitute a criticality in terms of saturation of the storage capacity. This is particularly evident when a specific level of stock is passed. Therefore, in [5] it is considered appropriate to model this dynamic through a storage cost curve which originally included a non-linearity from the exceeding of the critical stock level.

The linearity of the model is indeed guaranteed using a piece-wise linear curve that approximates the real cost curve. The indications about the threshold perceived by the waste company in relation to the customer service level can also be considered.

In the following, we introduce a mixed integer linear programming (MILP) model which defines the robust counterpart to the problem newly introduced in [5]. The basic notations that will be used in the MILP, such as parameters and indexes, are the following:

- $j \in \{1, \ldots, J\}$: index of the J sorting stages
- $p \in \{1, \ldots, P\}$: index of the P time-shifts
- T: time horizon partitioned in time shifts with $t \in \{1, \ldots, T\} = T_1 \cup \ldots \cup T_P$
- C: hourly cost of each operator
- σ_t: working hours for time t determined by the corresponding shift p
- $C_t = C * \sigma_t$: cost of each operator at time t
- f_j: set-up cost of sorting stage j
- a_t: quantity of material in kg unloaded from trucks at time t
- α_j: percentage of waste processed in stage $j-1$, received in input by buffer j
- S_j: maximum inventory capacity of the sorting stage buffer j
- LC_j: critical stock level threshold of buffer j
- ρ_j: fraction of material allowed to be left at buffer j at the end of time horizon
- K_j: single operator hourly production capacity [kg/h] of sorting stage j
- $SK_{j,t} = K_j * \sigma_t$: operator sorting capacity in sorting stage j, at time t
- M: maximum number of operators available in each time shift
- E_j: minimum number of operators to be employed in each time shift of stage j
- ∂h^i_j: slope of the i-th part of linearization of the buffer j stock cost curve

The nominal deterministic model consider the following variables.

- $x_{j,t} \in \mathbb{Z}^+$: operators employed in the sorting stage j at time t
- $u_{j,t} \in \mathbb{R}^+$: processed quantity at stage j at time t
- $y_{j,t} \in \{0, 1\}$: equal to 1 if stage j is activated at time t, 0 otherwise
- $I_{j,t} = I^{'}_{j,t} + I^{''}_{j,t} \geq 0$: stock level of material in buffer j at time t; for each stage j the corresponding $I'_{j,t}$ and $I''_{j,t}$ represent the inventory level before and after reaching the critical threshold respectively
- $w_{j,t} \in \{0, 1\}$: equal to 1 if $I^{''}_{j,t} > 0$, 0 otherwise. Indeed, these binary variables are used to model the piece-wise linear functions of the buffer stock costs.

We consider the set of parameters a_t, $t \in T$, that are subject to uncertainty taking values according to a symmetric distribution with mean equal to the nominal value a_t in the interval $[a_t - \hat{a}_t, a_t + \hat{a}_t]$. Indeed $\hat{a}_t$ is the maximum deviation of a_t. In order to meet the standard formulation of the nominal problem presented in [2], where parameters subject to uncertainties belong to inequality constraints only, the equality constraints of [5] regarding waste arrivals a_t are reformulated to turn them into inequality constraints. This is performed considering, for each period t, the sum of all the received and processed quantities of waste up to that period, as in constraints (5)–(8) of the formulation presented in this section. According to the

robust approach presented in [2], a parameter Γ_i is introduced for each constraint i holding one or more uncertainty coefficients. Γ_i is not necessarily integer and takes values in the interval $[0, |J_i|]$ where J_i is the set of the coefficients of constraint i being subject to uncertainty. The nominal problem presented in [5] presents only one set of T constraints considering the coefficients a_t and these are the ones reformulated as inequality constraints. Therefore we get $\Gamma \in R_+^T$, and because of this reformulation $|J_t| = t \ \forall t \in \{1, \ldots, T\}$. For each period t, Γ_t represents the number of coefficients that we consider as allowed to vary within their interval, ergo we consider nature behaving like only a subset of the coefficients will change with respect to their nominal value. Indeed, as affirmed in [2], it is unlikely that all $|J_t|$ will change; so the idea of conservative robustness is to be protected against all cases that up to $\lfloor \Gamma_t \rfloor$ of these coefficients are allowed to change, and one coefficient a_t changes by $(\Gamma_t - \lfloor \Gamma_t \rfloor)\hat{a}_t$. Note that when $\Gamma_t = 0 \ \forall t \in \{1, \ldots, T\}$ we get the nominal deterministic scenario, while setting $\Gamma_t = |J_t| = t \ \forall t \in \{1, \ldots, T\}$ represents solving the problem of the worst case scenario. It is clear then that by varying Γ the level of robustness can be flexibly adjusted against the level of conservatism of the solution. Considering the peculiar structure of the constraints including a_t is important: because of the telescopic expansion of each set J_t as t goes from 1 to T (i.e. $|J_{t+1}| = |J_t| + 1$), we consistently constraint Γ_t to be bigger or equal to Γ_{t-1}.

In the following, we present all the additional variables and parameters that are required to introduce the robustness protection functions presented in [2] and formulate the robust counterpart of the model presented in [5]:

- $\epsilon_t \in \mathbb{R}^+$: extra variables multiplying $a_t \ \forall t \in T$. These variables are introduced in order to have a variable multiplying the only set of parameters that are affected by uncertainty. These are indeed constrained to be equal to 1 $\forall t \in T$.
- $z_t \in \mathbb{R}^+$: variable resulted of duality within Bertsimas and Sim robustness theory; when multiplied by Γ_t provides its overall contribution to the protection function of constraint t.
- $p_{t,k} \in \mathbb{R}^+$: variable resulted of duality within Bertsimas and Sim robustness theory; provides its contribution to the protection function of constraint t with respect to the specific coefficient a_k.
- $s_t \in \mathbb{R}^+$: variable resulted of duality and Bertsimas and Sim robustness theory; multiplied by $\hat{a}_t$ sets the lower bound of the protection function contribution in each constraint t.
- Γ_t: parameter to adjust the level of robustness of each period t.

Considering a case study where $J = 2$ sorting stages, for the 1st sorting phase, $u_{1,t} \geq 0$ and $x_{1,t} \in \{0, 1\}$ represent the quantity of material to be selected and decision to activate the process respectively at time t. For the second sorting phase, $u_{2,t} \geq 0$ and $x_{2,t} \in \{0, 1\}$ represent the quantity of material to be selected and decision to activate the process respectively at time t. $I_{1,t}, I'_{1,t}, I''_{1,t} \geq 0$ are the inventory levels at 1st phase sorting buffer while $I_{2,t}, I'_{2,t}, I''_{2,t} \geq 0$ are inventory levels at second phase sorting buffer. As previously stated, w_1 and w_2 are used to

model the piece-wise linear functions of the buffer stock costs. In detail $w_1 = 0$ if $I'_{1,t} < LC$, 1 if $I'_{1,t} = LC$ and $I''_{1,t} > 0$; similarly $w_2 = 0$ if $I'_{2,t} < LC$, 1 if $I'_{2,t} = LC$ and $I''_{2,t} > 0$.

The model minimizes the sum of sorting and holding costs and is detailed as following:

$$\min Z = \sum_{j\in J}\sum_{t\in T} C_t x_{j,t} + \sum_{j\in J}\sum_{t\in T} f_j y_{j,t} + \sum_{j\in J}\sum_{t\in T}\left(\partial h_j^1 I'_{j,t} + \partial h_j^2 I''_{j,t}\right) \tag{1}$$

s.t.

$$E_j\ y_{j,t} \le x_{j,t} \le M\ y_{j,t} \qquad \forall j \in J, t \in T_p, p \in P \tag{2}$$

$$\sum_{j\in J} x_{j,t} \le M \qquad \forall t \in T \tag{3}$$

$$u_{j,t} \le SK_{j,t}\ x_{j,t} \qquad \forall j \in J, t \in T \tag{4}$$

$$I_{1,0} + \sum_{k=1}^{t} a_k \epsilon_k - \sum_{k=1}^{t} u_{1,k} + z_t \Gamma_t + \sum_{k=1}^{t} p_{t,k} \le S_1 \qquad \forall t \in T \tag{5}$$

$$I_{1,0} + \sum_{k=1}^{t} a_k \epsilon_k - \sum_{k=1}^{t} u_{1,k} \ge 0 \qquad \forall t \in T \tag{6}$$

$$I_{1,0} + \sum_{k=1}^{T} a_k \epsilon_k - \sum_{k=1}^{T} u_{1,k} + z_T \Gamma_T + \sum_{k=1}^{T} p_{T,k} \le \rho_1\ LC_1 \tag{7}$$

$$I_{1,t} = I_{1,0} + \sum_{k=1}^{t} a_k \epsilon_k - \sum_{k=1}^{t} u_{1,k} + z_t \Gamma_t + \sum_{k=1}^{t} p_{t,k} \qquad \forall t \in T \tag{8}$$

$$I_{j,t} = I_{j,t-1} - u_{j,t} + \alpha_j\ u_{j-1,t} \qquad \forall t \in T, j \in J \setminus 1 \tag{9}$$

$$I_{j,t} = I'_{j,t} + I''_{j,t} \qquad \forall j \in J, t \in T \tag{10}$$

$$LC_j\ w_{j,t} \le I'_{j,t} \le LC_j \qquad \forall j \in J, t \in T \tag{11}$$

$$0 \leq I_{j,t}^{''} \leq (S_j - LC_j)\, w_{j,t} \qquad \forall j \in J, t \in T \tag{12}$$

$$I_{j,T} \leq \rho_j\, LC_j \qquad \forall j \in J \setminus 1 \tag{13}$$

$$z_t + p_{t,k} \geq \widehat{a_t}\, s_t \qquad \forall t \in T, k \in \{0, \ldots, t\} \tag{14}$$

$$-s_t \leq \epsilon_t \leq s_t \qquad \forall t \in T \tag{15}$$

$$\epsilon_t = 1 \qquad \forall t \in T \tag{16}$$

$$x_{j,t} \in \mathbb{Z}^+ \qquad \forall j \in J, t \in T \tag{17}$$

$$u_{j,t} \in \mathbb{R}^+ \qquad \forall j \in J, t \in T \tag{18}$$

$$y_{j,t} \in \{0, 1\} \qquad \forall j \in J, t \in T \tag{19}$$

The objective function (1) defines the minimization of the sum of the three cost terms, which are sorting, setup, and inventory costs respectively. Equations (2) and (3) bounds the number of workers that can be assigned to each sorting station and to each time shift. Constraints (4) limit the quantity sorted $u_{j,t}$ to the sorting capacity dependent on the number of workers $x_{j,t}$. The following constraint sets (5)–(9) define and limit the inventories: constraint (5) defines the inventory for the first buffer, considering the cumulative inbound material a_t up to period t, the overall sorted material u_{1t} up to period t, and the uncertainties protection function made of the joint contribution of $z_t \Gamma_t$ and the sum of $p_{t,k}$ for $k \in \{1, \ldots, t\}$. Constraint (6) sets the lower bound of the inventory for each period and (7) imposes the maximum unsorted material allowed to be left at the end of the planning period for the first buffer, as constraint (13) does for all other subsequent buffers. Equality constraint (8) allows the inventory of the main buffer (i.e. buffer no. 1) to be considered in the corresponding piece-wise linear part of the cost function. Constraint (9) defines the inventory for the other buffers corresponding to $j > 1$. Indeed (9) outlines the waste flow across the sorting stages that follow one another: each subsequent inter-operational buffer j receives by the previous sorting stage $j - 1$ a quantity of waste equal to a α_j percentage of the waste processed in stage $j - 1$. Constraint sets (10), (11), and (12) define the piece-wise linear functions for inventories; in these constraints, maximum capacity level S_j and the critical stock level threshold LC_j are connected with the inventory levels through the variable $w_{j,t}$. Constraints (14) and (15) resulted from duality in [2] robustness theory; where (14) sets the lower bound of the protection function contribution in constraints (5) and (7).

3 Experimental Results

This section holds the main results from the studied scenarios described in the following. All instances are created by a real-world case study from a waste sorting plant located next to Rome, Italy. We solved each scenario by applying the Gurobi 9.0 solver to the MILP model. This has been performed in order to test the model response to different levels of robustness (i.e. different Γ selection), the corresponding price of robustness (i.e. the optimality reduction w.r.t the deterministic scenario) with respect to different weeks of scheduling time horizons. Figure 1 provides a first look at the model reaction to three different scenarios: deterministic case, an intermediate level of protection and the worst case scenario. It is evident that in the deterministic case the protection function value remain null for each period, almost like the first buffer stock level. Indeed the production marginal cost is less than the storage marginal cost resulting in the processing of waste as soon as arrives at the sorting facility. This is the reason why considering constraint (8) the protection function value equals for each period the first buffer stock level, and this applies for each protection scenario. Therefore we can attribute a cost to the protection function as the extra cost related to an higher level of stock in first buffer receiving the uncertain amount of waste. The robustness performance and the corresponding additional cost definitely depend on the protection strategy of choosing the vector $\Gamma \in R_+^T$ s.t. $\Gamma_t \geq \Gamma_{t-1} \forall t \in T$. In Fig. 1 the intermediate protection relies on a moderate still continuous increase of Γ_t. This approach represents a cumulative sum of protection over the risk considered across the time horizon. Dealing with robustness to reverse demand uncertainties in a scheduling problem setting is a suggestion to consider the seasonality of the stochastic behaviour of the coefficients when dealing with the strategy of choosing $\Gamma \in R_+^T$. In the considered real case

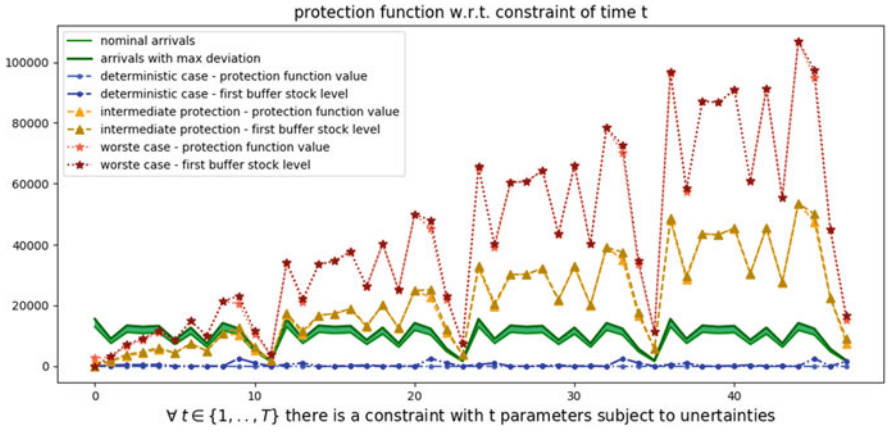


Fig. 1 Illustration of protection function and stock level evolution over three different uncertainties protection scenarios

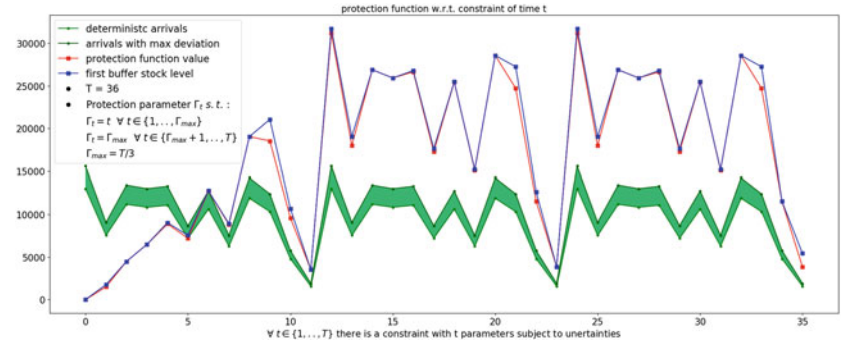


Fig. 2 Using demand period for Γ selection strategy

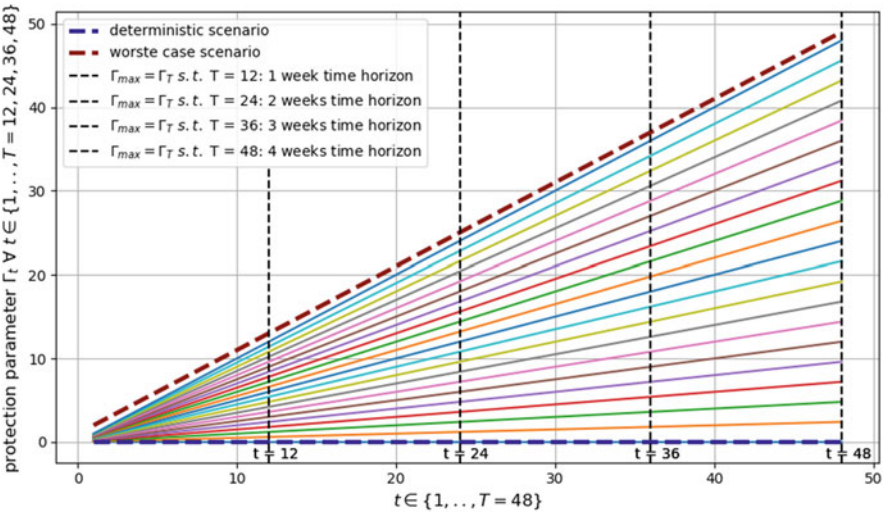


Fig. 3 protection magnitude scenarios: from deterministic to worst case

application the parameters a have a 1 week period (i.e. $t_{period} = 12$ when $P = 2$ working shift a day for six working days). Therefore a good approach is increasing Γ_t for $t \in \{1, \ldots, t_{period}\}$ and keeping the maximum Γ_{period} for the rest of the time horizon. Figure 2 shows an example with a 3 weeks time horizon (i.e. $T = 36$).

The price of robustness (the optimality reduction w.r.t. the nominal deterministic problem) is tested over twenty protection magnitudes with respect to different time horizons from one to 4 weeks. All Γ selections linearly increase with different slopes from minimum to maximum risk protections as shown in Fig. 3.

Results concerning the price of robustness are presented in Fig. 4. It is clear that the evolution of the price paid for risk protection remains reasonable and its evolution with respect to the protection scenarios strictly depend on the strategical selection of Γ. Indeed a linear evolution of the price is obtained with a linear expansion of Γ components.

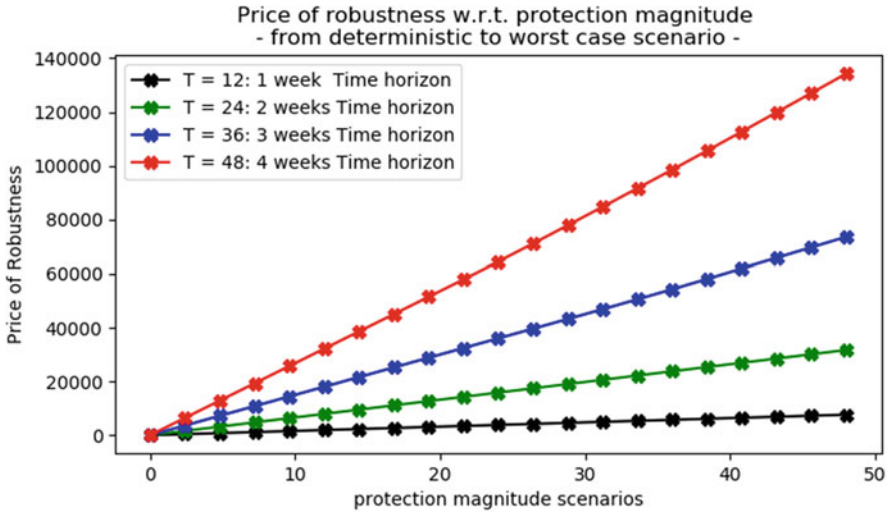


Fig. 4 Price of robustness results

4 Conclusions

We advanced a tuned version of the model presented in [5] with additional complexity due to the introduction of robustness on the most critical parameters values. The formulation keeps supporting all the original strategic decisions that are critical in the business considered. This robust counterpart showed a good adjustable protection capacity when used in a real-world application. Results concerning the controllable price of robustness in the considered case study are also encouraging. Indeed, for the company level of service, this economical and controllable improvement is highly remarkable, taking into account the low margin of the activity. Future works may consider to introduce more complexity in the formulation, such as considering production capacity dependent on the size of working teams.

Acknowledgments This work has been partially supported by EU POR-FESR program of LAZIO Region on "Circular Economy and Energy" through the project REMIND—REverse Manufacturing Innovation Decision system [Grant No. B86H18000160002] and POR-FESR on "Research Groups 2020" through the project PIPER—Piattaforma intelligente per lottimizzazione di operazioni di riciclo [Grant No. A0375-2020-36611, CUP B85F21001480002].

References

1. Aringhieri, R., et al.: A special vehicle routing problem arising in the optimization of waste disposal: a real case. Transp. Sci. **52**(2), 277–299 (2018)
2. Bertsimas, D., Sim, M.: The price of robustness. Oper. Res. **52**(1), 35–53 (2004)
3. Flavio Bonomo, F., et al. "A method for optimizing waste collection using mathematical programming: a Buenos Aires case study. Waste Manag. Res. **30**(3), 311–324 (2012)
4. Ghiani, G., et al.: Operations research in solid waste management: a survey of strategic and tactical issues. Comput. Oper. Res. **44**, 22–32 (2014)
5. Pinto, D.M., Stecca, G.: Optimal planning of waste sorting operations through mixed integer linear programming. In: C. Gentile, G. Stecca, P. Ventura (eds.) Graphs and Combinatorial Optimization: From Theory to Application. Springer AIRO Series, vol 5, pp. 307–320, 2020
6. Samanlioglu, F.: A multi-objective mathematical model for the industrial hazardous waste location-routing problem. Eur. J. Oper. Res. **226**(2), 332–340 (2013)
7. Stecca, G.: Electrical and electronic equipment recycling analyisis and simulation for urban reverse logistics services. In: International Symposium on Flexible Automation Awaji-Island, Hyogo, Japan, 14–16 July, pp. 1–6. ISFA (2014)

Solution Approaches for the Capacitated Scheduling Problem with Conflict Jobs

Emanuele Tresoldi

Abstract In this paper we present a new arc-based mathematical formulation and a heuristic algorithm for the capacitated scheduling problem with conflict jobs. The effectiveness of our approaches is demonstrated through extensive computational experiments. The results demonstrate that our formulation outperforms existing models proposed in the literature.

Keywords Scheduling with conflicts · Parallel machines · Arc-Flow formulation

1 Introduction

The Capacitated Scheduling Problem with Conflicts Jobs (CSPCJ) requires to find a feasible schedule, on a set of parallel identical machine without preemption, that maximizes the total weighted value of jobs completed before a common deadline. The schedule is subject to conflict constraints limiting the set of jobs that can be processed concurrently. Using the standard three-field notation for scheduling problems introduced in [4] the CSPCJ can then be classified as follows. The first field is a single character identifying the type of machines considered, in CSPCJ is P since it is defined on parallel identical machines. The second, describing constraints on jobs, is $d_j = d$. In the three field notation d identifies the due date requirements. Given the set of jobs $\mathcal{J}$ each job $j \in \mathcal{J}$ has the same due date $d_j = d$. The last field describe the objective function. In CSPCJ is $\sum w_j U_j$ meaning that it requires the minimization of a weighted sum function where w_j is the weight of a job and U_j is a unit function equal to 1 if job j is completed before d and 0 otherwise. The complete problem can then be defined as $P|d_j = d|\sum w_j U_j$ with the addition of conflict constraints.

E. Tresoldi (✉)
Dipartimento di Informatica, Università degli Studi di Milano, Milan, Italy
e-mail: emanuele.tresoldi@unimi.it

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_11

CSPCJ shares many features with other well studied scheduling problems such as *the mutual exclusion scheduling* [1], *the parallel machine scheduling with conflict graph* [6], *the scheduling with agreement graph* [2], *the flow shop scheduling with conflict graph* [12] and *the open shop scheduling with conflict graph* [11]. On all these problems a set of jobs has to be scheduled on a set of parallel processors subject to constraints limiting/forcing the concurrent execution of jobs. Two peculiar features differentiate CSPCJ from all these problems: the objective function and the machine capacity/common deadline for all jobs. Indeed, in the previously reported problems the objective function is usually the makespan minimization and none of them take into consideration the capacity of the machines or any common due time for the jobs. It is worth noting that, in literature, there is another problem that is identified by a very similar name: *the scheduling with conflicts* [7]. However, in this case if two jobs are in conflict they cannot be scheduled on the same machine. The structure of this problem is very different from the CSPCJ.

CSPCJ has received very little attention in literature; to the best of our knowledge there is only one paper [5] that specifically deals with CSPCJ. In this paper the authors introduce the problem, describe a real-world application in the field of wireless network management, prove the NP-hardness of CSPCJ and identify polynomially solvable special cases. Moreover, they provide two different Mixed-Integer Linear Programming (MILP) models for CSPCJ. These models, called F1 and F2 by the authors, have been tested on a small set of limited size instances (up to 16 jobs). The experimental results show the limitations of the proposed models. In fact, neither F1 nor F2 is able to solve all instances and many cases show very large optimality gap (up to 80%) after one hour of computation.

In order to find better solutions for CSPCJ, in this paper we introduce a new mathematical formulation for the problem (Sect. 2.1) and a heuristic algorithm to quickly provide good-quality feasible solutions (Sect. 3). To show the effectiveness of our approaches we first compare them with the state of the art and then we test them on large size instances (Sect. 4). Finally, we close the paper drawing our conclusions (Sect. 5).

2 Mathematical Formulations

The CSPCJ can be formally defined as follows. A set $\mathcal{J}$ of n jobs $\mathcal{J} = \{1 \dots n\}$ and a set $\mathcal{I}$ of parallel identical machines $\mathcal{I} = \{1 \dots m\}$ are given. Each job $j \in \mathcal{J}$ is characterized by an integer processing time p_j and an integer weight w_j while all jobs share a common deadline T. Moreover, given a set $\mathcal{V} \subseteq \mathcal{J}$ of all jobs with at least one conflict and a set $\mathcal{E}$ of all pairs of conflicting jobs the conflict requirements are described by means of an undirected graph $G(\mathcal{V}, \mathcal{E})$. In details, an edge $(j, h) \in \mathcal{E}$ identifies two jobs j and $h \in \mathcal{V}$ that cannot be scheduled concurrently. The processing of job j must start after the ending of job h or vice-versa. In other words, j and h must be processed in disjoint time intervals. The goal of CSPCJ is to build a

feasible schedule respecting all conflict constraints and maximizing the sum of the weights of the jobs that are completed before the deadline.

2.1 Arc-Time Formulation

In this section we introduce a new formulation for CSPCJ inspired by arc-flow models for different scheduling problems such as [9] and [8]. Our formulation, however, is not a straight adaptation of arc-flow models presented in literature. Indeed, since in the optimal solution of CSPCJ there may be a waiting period between the processing of two consecutive jobs, in our model the problem is not described as a flow. However, we sill have arcs in our formulation. They always connects two points in time: the beginning and the end of the processing of a job. For this reason we called this formulation arc-time formulation (AT).

Formulation AT is defined over the set $\mathcal{T} = \{1, 2, \dots T\}$ containing all time instants in which the processing of a job can either start or end. It is worth noting that, since all processing times are integer, this discretization does not affect the optimality of the solutions found. This formulation requires, for each job $j \in \mathcal{J}$, the definition of two subsets. The first subset $\mathcal{E}_j \subset \mathcal{E}$ containing all jobs conflicting with j, the other $\mathcal{T}_j \subset \mathcal{T}$ contains all time instants t such that $t + p_j \leq T$.

AT makes use of a single family of binary variables x_j^t ($\forall j \in \mathcal{J}, t \in \mathcal{T}_j$) equal to 1 if the processing of job j starts at t and 0 otherwise. They represent arcs from time instants t to $t + p_j$. Formulation AT then reads as follows:

$$\max \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}_j} w_j x_j^t \tag{1a}$$

$$\sum_{t \in \mathcal{T}_j} x_j^t \leq 1 \qquad \forall j \in \mathcal{J} \tag{1b}$$

$$\sum_{j \in \mathcal{J}} \sum_{\substack{t' \in \mathcal{T}_j: \\ t' \leq t < t' + p_j}} x_j^{t'} \leq |I| \qquad \forall t \in \mathcal{T} \tag{1c}$$

$$\sum_{\substack{t' \in \mathcal{T}_j: \\ t' \leq t < t' + p_j}} |I| x_j^{t'} + \sum_{h \in \mathcal{E}_j} \sum_{\substack{t' \in \mathcal{T}_h: \\ t' \leq t < t' + p_h}} x_h^{t'} \leq |I| \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \tag{1d}$$

$$x_j^t \in \{0, 1\} \qquad \forall j \in \mathcal{J}, t \in \mathcal{T} \tag{1e}$$

The objective function (1a) calls for the maximization of the total weight of the jobs completed within the deadline. Constraints (1b) ensure that a job cannot be processed more than once. Inequalities (1c) model the concurrent allocation of jobs to machines. It imposes that, in each time instant $t \in \mathcal{T}$, the number of jobs

that are actively processed is no larger than the number of available machines $|I|$. Finally, the last family of constraints (1d) addresses the conflict requirements. These constraints are violated if two conflicting jobs are processed in the same time instant $t \in \mathcal{T}$. It is worth noting that another way of imposing conflicts requirements in AT is with inequalities:

$$\sum_{\substack{t' \in \mathcal{T}_j: \\ t' \leq t < t' + p_j}} x_j^{t'} + \sum_{\substack{t' \in \mathcal{T}_h: \\ t' \leq t < t' + p_h}} x_h^{t'} \leq 1 \quad \forall (j,h) \in \mathcal{E}, t \in \mathcal{T} \tag{2}$$

These inequalities dominate (1d). However, we established in a preliminary testing phase that, due to the cardinality of (2), constraints (1d) provide better computational performances.

The AT formulation includes, as in all arc-flow inspired models, a pseudo-polynomial number of variables that depends on the cardinality of set $\mathcal{J}$ and parameter T. Usually, in these models, different reduction and symmetry breaking techniques are employed to discard redundant variables (see [3]). Unfortunately, in AT no variable can be discarded. Indeed since CSPCJ includes conflicts and potential waiting times none of the methods presented in the literature can be applied to our model.

Additional Inequalities Let be $\bar{\mathcal{Q}}$ the set of all possible cliques in the conflict graph $G(\mathcal{V}, \mathcal{E})$ and let $\mathcal{V}_q \subset \mathcal{V}$ be the set of nodes defining clique $q \in \bar{\mathcal{Q}}$ then feasible inequalities for AT are defined as follows:

$$\sum_{j \in \mathcal{V}_q} x_j^t \leq 1 \quad \forall q \in \bar{\mathcal{Q}}, t \in \mathcal{T} \tag{3}$$

A clique in the conflict graph represent a set of jobs that cannot run concurrently. Therefore, at any time instant at most one of them can be processed.

3 Heuristic Algorithm

In this section we present a simple heuristic algorithm (HE) for CSPCJ, it combines randomization, greedy construction and local search improvement. An overview of the procedure is shown in Algorithm 1. The core of the algorithm is based on three sub-procedures that iteratively build and improve the solution. These procedures make use of several data-structures. In details, an $|\mathcal{I}|$ by T matrix of time instants U representing the allocation of jobs to machines; e.g. $U_i^t = j$ if job j is processed

by machine i starting at time t. A ordered list of the jobs $\bar{J}$ and a set S of triplets (i, j, t) representing the solution (e.g. job j is allocated to machine i starting from time instant t).

The core algorithm (rows 3 and 10) starts form an empty solution in which no job is scheduled (rows 3 and 4) and, given the list of jobs randomly ordered $\bar{J}$ (row 5), builds a better solution iterating the three sub-procedures. At first it allocates jobs to machines in a greedy fashion (row 7). When no other job can be scheduled a local search procedure is executed trying to swap jobs in the solution with more profitable jobs not already allocated (row 8). Finally the solution is compacted moving all jobs as far as possible form the deadline (row 9). Each sub-procedure takes as input the output of the previous one. Every time the solution is either improved or shrunk these sub-procedures are repeated. The core algorithm is executed L times, every time $\bar{J}$ is ordered in a different way. At the end the best solution obtained is returned.

Algorithm 1 Heuristic procedure—overview

Input: $\mathcal{I}$ machines, $\bar{J}$ jobs, U covering matrix, T deadline, p processing times, w weights, G conflict graph, L max iterations.
Output: s^* value best solution, S^* triplets (i, j, t) defining best solution.

```
1  s* ← 0; S* ← ∅;
2  while l < L do
3      s ← 0; S ← ∅;
4      U_i^t ← ∅ ∀i ∈ I, t ∈ {1 ... T};
5      J̄ ← shuffle(J̄);
6      do
7          B_g, s, S, U ← greedyInsert(s, S, I, J̄, U, T, p, q, G);
8          B_s, s, S, U ← localSearch(s, S, I, U, T, p, q, G);
9          B_c, S, U ← compact(S, U, G);
10     while B_g ∨ B_s ∨ B_c;
11     if s > s* then
12         s* ← s S* ← S;
13     l ← l + 1;
14 return s*, S*
```

The pseudo-code of the greedy insertion procedure is reported in Algorithm 2. This procedure scans the matrix U looking for unused time instants (rows 2 and 3). Once an available time instant t in a machine i is located (row 4), the procedure, following the order of $\bar{J}$, try to insert new jobs into the solution. If the processing time of the job fits into the available time instants (*checkTime*, row 6) and does not have any conflict with pre-allocated jobs in the same time instants (*checkConflicts*, row 6) then the job is allocated on machine i starting at time t. The matrix U and the solution are updated (row 9) and the procedure continue scanning for other available time instants.

Algorithm 2 Heuristic procedure—greedy insertion

Input: s value ini. solution, S ini. solution, $\mathcal{I}$ machines, $\bar{J}$ jobs, U matrix, T deadline, p processing times, w weights, G conflict graph.
Output: B_g boolean, s^* value solution, S^* solution, U^* matrix of solution.

```
1  s* ← s; S* ← S; U* ← U; B ← FALSE;
2  for t ← 0 to T do
3      forall i ∈ I do
4          if U_i^{*t'} = ∅ then
5              forall j ∈ J̄ \ S* do
6                  if checkTime(t, i, j, U) ∧ checkConflicts(t, j, U, G) then
7                      for t' ← t to t + p_j do
8                          U_i^{*t'} ← j;
9                      s* ← s* + w_j; S* ← S* ⋃ (i, j, t); B ← TRUE;
10                     break;
11 return B, s*, S*, U*
```

The second sub-procedure is a local-search algorithm, the pseudo-code is shown in Algorithm 3. This procedure starts form a feasible solution $S^* = S$ and tries to exchange a job $\in S^*$ with a more profitable job $\notin S^*$ following the order described by $\bar{J}$. Two jobs j and j' can be exchanged if j' fits in the hole that would be created by removing j (row 5) and if inserting j' in its place does not create any conflict (row 6). The exchange is realized if either j' is more profitable than j or j' is as profitable as j and the processing time of j' is shorter than the processing time of j (row 4). Once an exchange is performed then the matrix U and the solution S^* are updated and the procedure moves to the next job $\in S^*$.

Algorithm 3 Heuristic procedure—local search

Input: s value ini. solution, S ini.l solution, $\mathcal{I}$ machines, $\bar{J}$ jobs, U matrix, T deadline, p processing times, w weights, G conflict graph.
Output: B_g boolean, s^* value solution, S^* solution, U^* matrix of solution.

```
1  s* ← s; S* ← S; U* ← U; B ← FALSE;
2  forall (i, j, t) ∈ S* do
3      forall j' ∈ J̄ \ S* do
4          if w_{j'} > w_j ∨ (w_{j'} = w_j ∧ p_{j'} < p_j) then
5              t' ← checkHole(t, i, j, j', U);
6              if t' > 0 ∧ checkConflicts(t, j, U, G) then
7                  for t' ← t to t + p_j do
8                      U_i^{*t'} ← ∅;
9                  for t' ← t̄ to t̄ + p_{j'} do
10                     U_i^{*t'} ← j';
11                 s* ← s* − w_j + w_{j'}; S* ← S* ⋃ (i, j', t') \ (i, j, t); B ← TRUE;
12                 break;
13 return B, s*, S*, U*
```

Finally, since the local search procedure can generate unnecessary holes (e.g. exchanging a large job with a smaller one, conflicts changed, etc.) a third procedure (*compact*) is used in order to compact the schedule as much as possible. In details,

all jobs, following the order described by $\bar{J}$ are moved as far away as possible from the deadline (toward $t = 1$). This may involve the relocation of a job on a different machine if a suitable set of time instants is available.

It is worth noting that the complete algorithm (Algorithm 1) is easily parallelizable running each iteration on a separate process.

4 Experimental Evaluation

In order to assess the performances of our approaches we executed two different experimental tests on two separate sets of instances. The first set is composed of 144 random instances generated to replicate the test-set described in [5]. They are characterized by a number of jobs $|\mathcal{J}| = \{12, 16\}$, a number of machines $|\mathcal{I}| = \{2, 4\}$, a deadline $T = \{15, 30, 60\}$, a ratio between the number of conflicts and the number of jobs equal to $C = \{0.2, 0.5, 1\}$ and two settings for the generation of processing times: a random integer from $\lceil T/4 \rceil$ to $\lceil T/2 \rceil$ (setting 1) or from 1 to $\lceil T/2 \rceil$ (setting 2). For each combination of jobs, machines, deadline, conflict ratio and processing time settings two instances with random weights $\in \{1 \ldots 5\}$ are generated. This test set is used to compare the AT model (Sect. 2.1) and the heuristic algorithm (Sect. 3) with the models F1 and F2 described in [5]. The second set contains 3840 larger size instances generated with $|\mathcal{J}| = \{24, 32, 48, 64\}$, $|\mathcal{I}| = \{2, 4, 6, 8\}$, $T = \{30, 60, 120, 180\}$, $C = \{0.2, 0.5, 1\}$ and the same two setting for processing times. This set has been generated to provide more challenging instances and to test the limits of our approaches. Both datasets and additional small instances are available on *Open Science Framework* [13].

All models (F1, F2 and AT) and the heuristic algorithm (HE) have been implemented in `C++`. CPLEX 20.1 `C++ API` has been used for all three models.

We include in model AT additional inequalities (3). Before the model is solved we look for maximal cliques in the conflict graph; all cliques found are then converted into constraints as described in Sect. 2.1. This process is carried out using the maximum clique algorithm described in [10], all additional constraints are then added to AT as cuts.

The experiments were carried out on a 64-bit Windows machine, with Intel i7-6700K processor clocked at 4.00 GHz and 16 GB RAM. CPLEX is set to default configuration and up to 8 thread are used for parallel optimization. The heuristic algorithm runs on a single thread instead. We set a time-limit of 3600 seconds for every instance on all models while a limit of 100 iterations for HE has been imposed (parameter L in Algorithm 1). The computational time for the maximum clique algorithm is included in the time-limit.

Comparison with the State of the Art

To compare our approaches with the state of the art we run formulations F1, F2, AT and our heuristic algorithm on the first data-set. The results of this comparison, aggregated by jobs, machines, deadline and conflict ratio are shown in Table 1.

Table 1 Comparison F1 vs F2 vs AT vs HE

Instances				F1			F2			AT			HE		
$\|\mathcal{J}\|$	$\|\mathcal{I}\|$	T	C	Time	Gap	Op	Time	Gap	Op	Time	Gap	Op	Time	Gap	Op
12	2	0.2	15	1803.87	13.55	2	39.48	0.00	4	0.11	0.00	4	0.01	0.00	4
12	2	0.2	30	1807.84	21.50	2	20.16	0.00	4	0.17	0.00	4	0.02	0.00	4
12	2	0.2	60	1804.76	15.25	2	13.25	0.00	4	0.45	0.00	4	0.02	0.00	4
12	2	0.5	15	1806.10	30.05	2	10.02	0.00	4	0.13	0.00	4	0.01	0.00	4
12	2	0.5	30	1804.20	12.30	2	25.53	0.00	4	0.21	0.00	4	0.02	0.00	4
12	2	0.5	60	1803.97	5.20	2	7.44	0.00	4	0.68	0.00	4	0.02	0.00	4
12	2	1.0	15	1803.82	10.75	2	2.07	0.00	4	0.12	0.00	4	0.02	0.00	4
12	2	1.0	30	1804.04	14.75	2	4.89	0.00	4	0.26	0.00	4	0.02	0.00	4
12	2	1.0	60	1803.58	22.80	2	3.11	0.00	4	1.66	0.00	4	0.03	0.00	4
12	4	0.2	15	1309.54	3.10	3	3.82	0.00	4	0.10	0.00	4	0.03	0.00	4
12	4	0.2	30	1425.45	2.80	3	32.85	0.00	4	0.18	0.00	4	0.06	0.00	4
12	4	0.2	60	376.36	0.00	4	3.90	0.00	4	0.41	0.00	4	0.07	0.00	4
12	4	0.5	15	1026.94	2.70	3	1.42	0.00	4	0.12	0.00	4	0.03	0.00	4
12	4	0.5	30	1021.42	3.30	3	0.92	0.00	4	0.24	0.00	4	0.13	0.00	4
12	4	0.5	60	999.59	2.60	3	0.71	0.00	4	0.56	0.00	4	0.08	0.00	4
12	4	1.0	15	219.50	0.00	4	0.26	0.00	4	0.12	0.00	4	0.04	0.00	4
12	4	1.0	30	178.06	0.00	4	0.25	0.00	4	0.23	0.00	4	0.06	0.00	4
12	4	1.0	60	14.54	0.00	4	0.31	0.00	4	1.00	0.00	4	0.10	0.00	4
16	2	0.2	15	1806.27	46.45	2	239.10	0.00	4	0.12	0.00	4	0.02	0.00	4
16	2	0.2	30	1807.73	63.35	2	298.94	0.00	4	0.28	0.00	4	0.02	0.00	4
16	2	0.2	60	1821.52	49.40	2	2007.29	27.30	3	0.87	0.00	4	0.02	0.00	4
16	2	0.5	15	1813.57	46.95	2	636.98	0.00	4	0.17	0.00	4	0.01	0.00	4
16	2	0.5	30	1818.91	25.60	2	1252.75	5.10	3	0.30	0.00	4	0.02	0.00	4

16	2	0.5	60	1807.69	46.35	2	666.52	0.00	4	1.31	0.00	4	0.02	0.00	4
16	2	1.0	15	1813.40	26.70	2	1261.08	6.70	3	0.14	0.00	4	0.01	0.00	4
16	2	1.0	30	1807.00	24.70	2	258.68	0.00	4	0.45	0.00	4	0.02	0.00	4
16	2	1.0	60	1807.52	40.25	2	88.36	0.00	4	2.13	0.00	4	0.02	0.00	4
16	4	0.2	15	3605.96	17.50	0	1737.12	2.00	3	0.11	0.00	4	0.04	0.00	4
16	4	0.2	30	3603.97	9.80	0	1766.71	2.90	3	0.21	0.00	4	0.05	0.00	4
16	4	0.2	60	2703.49	14.83	1	1183.90	12.80	3	0.57	0.00	4	0.16	0.00	4
16	4	0.5	15	3605.09	11.78	0	1966.66	5.60	2	0.16	0.00	4	0.04	0.00	4
16	4	0.5	30	2706.50	12.83	1	346.45	0.00	4	0.30	0.00	4	0.06	0.00	4
16	4	0.5	60	2703.55	11.27	1	1894.34	4.30	2	0.90	0.00	4	0.07	0.00	4
16	4	1.0	15	3603.72	13.48	0	183.75	0.00	4	0.17	0.00	4	0.04	0.00	4
16	4	1.0	30	2704.55	12.97	1	131.33	0.00	4	0.35	0.00	4	0.06	0.00	4
16	4	1.0	60	2710.87	9.57	1	95.62	0.00	4	1.93	0.00	4	0.10	0.00	4

Columns "Time" report the average computational time, over the four instances of the same class, in seconds. Columns "Gap" show the average MIP gap after one hour of computation, this gap is computed considering only instances that are not solved to the optimality. For HE the gap is computed with respect to the optimal solution. Columns "Op" contain the number of instances solved to the optimality in each class.

The performances of F1 and F2 are in line with the results in [5]; neither F1 nor F2 could find the optimal solution on all 144 instances. In details, formulation F2 is more efficient and is able to solve 134 instances while only 72 are solved by F1. The average computational time is about 449 s for formulation F2 and 1862 second for F1. On instances not solved to the optimality the average MIP gap for F1 is 20.01%, 46 instances report gap larger than 10%. With formulation F2 the MIP gap at the end of the computation is about 7.66% and only two instances shows a gap $> 10\%$. However, F1 provides a better relaxation, at root node the F1 MIP gap is, on average, about 25.78% while the F2 gap is 55.39% and on 32 instances (all with 2 machines) this results in F1 outperforming F2 by more than 70%.

Formulation AT outperforms both F1 and F2. Indeed AT is able to solve all instances to the optimality in about 1 second. The linear relaxation of AT provides a very small gap, on average about 0.34%, and no branching is required to achieve the optimal solution. The maximum clique algorithm used to find additional inequalities takes only a fraction of a second. Finally, the heuristic algorithm is very effective on this set of small instances. It reached the optimal solution on all instances in about a tenth of a second. Considered the performances showed by F1 and F2 we decided not to test them on larger instances.

Results on Larger Instances

The second set of instances is used to test the efficiency of AT and HE on larger problems. Average results, aggregated by $|\mathcal{J}|$ and T are reported in Table 2, each class contains 240 instances. For AT, columns "GapR" and "OpR" show, respectively, the average gap, for instances not solved, and the number of instances solved at root node. Columns "GapF", "OptF", "BBN" and "Time" report, the final gap, the number of instances solved within the time-limit, the number of branch-and-bound nodes explored and the total computational time in seconds. For HE, we report the average gap with respect to the best known solution "GapOp", the execution time "Time" and the number of instances in which the best known solution solution has been reached "Op".

Formulation AT is able to obtain the optimal solution for all instances but one, the average computational time is 35 seconds. About 80% of the instances are solved at the root node and the average MIP gap at the root node for the other ones is 1.64%. In these instances, on average, 19487 nodes are explored before optimality is proven. In the only instance not solved to the optimality the final MIP gap is 0.86%. The maximum clique algorithm used to compute additional inequalities runs in less than a second.

Table 2 Results AT and HE on new instances

Instances		AT						HE		
$\lvert\mathcal{J}\rvert$	T	GapR	OpR	GapF	BBN	Time	OpF	GapOp	Time	Op
24	30	2.85	215	0.00	0.00	0.82	240	2.18	0.08	172
24	60	2.28	206	0.00	75.27	4.12	240	2.32	0.11	153
24	120	2.21	192	0.00	145.78	15.76	240	2.09	0.19	158
24	180	2.31	190	0.00	417.15	36.85	240	2.40	0.28	159
32	30	2.53	198	0.00	0.00	2.56	240	1.79	0.10	143
32	60	1.67	197	0.00	703.41	8.44	240	1.93	0.14	120
32	120	1.41	186	0.00	87.49	29.85	240	2.07	0.22	114
32	180	1.46	189	0.00	31.02	66.77	240	2.07	0.28	113
48	30	1.12	203	0.00	54.97	3.81	240	1.48	0.14	110
48	60	1.68	186	0.00	19.93	5.87	240	1.83	0.19	97
48	120	1.43	195	0.00	2087.18	43.24	240	1.91	0.27	92
48	180	1.28	192	0.00	67.46	98.28	240	1.98	0.35	92
64	30	0.43	216	0.00	0.00	2.84	240	1.57	0.17	96
64	60	1.50	179	0.00	31.80	9.58	240	1.88	0.22	82
64	120	1.28	197	0.00	729.35	59.19	240	1.83	0.33	76
64	180	1.16	184	0.86	258.58	183.14	239	1.85	0.42	73

Algorithm HE reaches the best known solution in 1851 instances (48%). The gap from the best known solution in the other instances is on average 1.91%, only one instance shows a gap larger than 10% (11.76%) and a total of 15 instances report gaps larger than 5%. The overall average computational time is about 0.21 s.

5 Conclusions

The experimental results demonstrate the effectiveness of the proposed approaches. In particular, the AT model proved to be vastly superior to F1 and F2. In order to find the first instance not solved to the optimality we have to increase by four times the number of jobs and by three times the deadline. Moreover, we demonstrated that our simple heuristic algorithm is able to find very good solutions in a fraction of a second. It is worth noting that given the parallelizable nature of the algorithm it could be used to tackle much larger instances in a very short computational time.

Acknowledgments This work has been partially funded by Regione Lombardia, grant agreement n. E97F17000000009, Project AD-COM.

References

1. Baker, B.S., Coffman, E.G.: Mutual exclusion scheduling. Theor. Comput. Sci. **162**(2), 225–243 (1996)
2. Bendraouche, M., Boudhar, M.: Scheduling jobs on identical machines with agreement graph. Comput. Oper. Res. **39**(2), 382–390 (2012)
3. de Carvalho, J.V: Exact solution of bin-packing problems using column generation and branch-and-bound. Ann. Oper. Res. **86**(0), 629–659 (1999)
4. Graham, R., Lawler, E., Lenstra, J., Kan, A.: Optimization and approximation in deterministic sequencing and scheduling: a survey. In: Hammer, P., Johnson, E., Korte, B. (eds.) Discrete Optimization II, Annals of Discrete Mathematics, vol. 5, pp. 287–326. Elsevier, New York (1979)
5. Ha, M., Vu, D., Zinder, Y., Thanh, T.: On the capacitated scheduling problem with conflict jobs. In: 11th International Conference on Knowledge and Systems Engineering (KSE), pp. 1–5 (2019)
6. Hong, H.C., Lin, B.M.: Parallel dedicated machine scheduling with conflict graphs. Comput. Ind. Eng. **124**, 316–321 (2018)
7. Kowalczyk, D., Leus, R.: An exact algorithm for parallel machine scheduling with conflicts. J. Sched. **20**(4), 355–372 (2017)
8. Kramer, A., Dell'Amico, M., Iori, M.: Enhanced arc-flow formulations to minimize weighted completion time on identical parallel machines. Eur. J. Oper. Res. **275**(1), 67–79 (2019)
9. Mrad, M., Souayah, N.: An arc-flow model for the makespan minimization problem on identical parallel machines. IEEE Access **PP**, 5300–5307 (2018)
10. Pattabiraman, B., Patwary, M.M.A., Gebremedhin, A.H., Liao, W., Choudhary, A.N.: Fast algorithms for the maximum clique problem on massive graphs with applications to overlapping community detection. CoRR abs/1411.7460 (2014)
11. Tellache, N.E.H., Boudhar, M.: Open shop scheduling problems with conflict graphs. Discrete Appl. Math. **227**, 103–120 (2017)
12. Tellache N.E.H., Boudhar, M.: Flow shop scheduling problem with conflict graphs. Ann. Oper. Res. **261**, 339–363 (2018). https://doi.org/10.1007/s10479-017-2560-x
13. Tresoldi, E.: Capacitated scheduling problem with conflict jobs - dataset. https://osf.io/7xks4/. Accessed 25 Jan 2021

Part IV
Transportation and Logistics

A Decision Model for Enhancing Driving Security

Mauro Maria Baldi, Nicola Cilli, Enza Messina, and Fabio Tango

Abstract Driving is a complex activity which requires constant care and attention. Intelligent Advance Driver Assistance Systems (ADAS) can improve vehicle control performance and, thus, drivers and passengers safety. In particular, identification and prediction of driving intention can provide prompt information to drivers and vehicles in their vicinity that are fundamental for avoiding collisions. In this paper, we propose a lane change prediction model based on machine learning able to distinguish between left and right lane changes, a distinction that becomes particularly important when driving in a highway. Models have been trained and validated using a real dataset gathered online by using a high-tech demonstrator vehicle provided by Centro Ricerche Fiat (i.e., Fiat Research Center). Data, which refer to real driving conditions on a highway, have been collected by monitoring different drivers showing different behaviors. We address the problem of unbalanced data, typical of real data sets, and propose two prediction models based on Support Vector Machines and Random Forests. The results of our computational experiments show the validity of the approach with respect to state of the art models, both in terms of prediction accuracy and prediction time.

Keywords Advance driver assistance systems · Decision models · Machine learning · Interaction Human-automation

M. M. Baldi (✉) · N. Cilli · E. Messina
Università degli Studi di Milano-Bicocca, Milano, Italy
e-mail: mauro.baldi@unimib.it; n.cilli@campus.unimib.it; enza.messina@unimib.it

F. Tango
Centro Ricerche Fiat, Orbassano, Italy
e-mail: fabio.tango@crf.it

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_12

1 Introduction

Driving is a complex and dangerous activity because even a little distraction can have serious consequences. According to recent studies [1], car is the predominant mode of transport reaching in Europe a global percentage of use of 82.9%. Similar studies [2] claim that in 2018 in the European Community 23,000 people died in road accidents in contrast to 1666 significant railway accidents with a high impact of transport safety. It is therefore of upmost importance to develope new technologies for improving driving safety. In order to enhance the driving experience from a safety point of view, the so-called Advance Driver Assistance Systems (ADAS) have been developed [3]. Lane change is one of the most dangerous maneuvers and it is precisely in this context that a decision model can be useful to improve safety in carrying out this maneuver.

This paper is an extension of a work [4] rooted in the European project DESERVE [5] in cooperation with Centro Ricerche Fiat (CRF) [6] (i.e., Fiat Research Center) and whose goal is to increase driving security through the development of an embedded ADAS. In contrast to the preliminary work in [4] which showed the results over a preliminary (and small) dataset, in this paper we present new results over the real dataset created during the DESERVE project. This is an important contribution because most of the articles in the literature present works either based on data obtained by simulation or on real data but produced by considering only a small number of drivers. This is the case, for example, of all those works based on the popular NGSIM dataset [7], concerning a short section of two American highways located in the Los Angeles and San Francisco areas, California, USA. Some limitations of this dataset have been discussed by a number of researchers [8–10] that raised important issues rooted in unrealistic relationships in the data. Relevant efforts have been made to correct these flaws [11, 12]. Nevertheless, as Coifman and Li [13] state, "the NGSIM errors are beyond anything that could be corrected strictly through cleaning or interpolation of the reported NGSIM data". Instead in this study, we consider data gathered online from a considerable number of drivers. Driving tests of a number of drivers for a reasonable long distance allowed the construction of a real dataset on which we trained a decision model to predict a lane change maneuver. As the main maneuver when driving is lane keeping, the resulting dataset is heavily unbalanced. Indeed, although a driver can make a lot of overtaking, he or she will mainly be in a condition of lane keeping. For this reason, we have also used the SMOTETomek algorithm [14, 15] which allows to effectively deal with an unbalanced dataset, favoring the separability of the classes.

When driving in a highway, the distinction between left and right lane change becomes fundamental. In fact, as we noticed from our data, a right lane change is usually a calmer maneuver than a left lane change. For this reason, we learned models capable of recognising not only lane keeping and lane change maneuvers, as most state of the art works do, but also able to distinguish between left lane change before the passing phase and right lane change, occurring when returning to

the original lane. Moreover, it is important to note that in order to assist the drivers, ADAS systems should predict lane change maneuvers early enough to warn the driver on time. In light of this principle, we computed the average prediction time for a lane change, which ranges between 1 and 2 s. The classification accuracy of our models ranges from 87% to 90%; a result perfectly in line with other works in the literature.

The paper is organized as follows: in Sect. 2 we provide a literature review on lane change maneuvers prediction. Section 3 introduces our case study in detail from driving tests up to the preprocessing phase. Computational results are shown in Sect. 4 and we conclude in Sect. 5.

2 Literature Review

A rich literature is available on machine learning approaches for lane change detection. Some works are based on Hidden Markov Probabilistic Models [16–19]. In particular, dynamic belief networks, a general case of Hidden Markov Models, have been proposed by Dagli et al. [20]. Despite probabilistic models, also Support Vector Machines (SVM) have been employed at this purpose [18, 21–24]. Recently, Li et al. [25] applied a generalization of SVM known as Support Vector Regression. Cognitive models can be found in [26, 27], while McCall et al. [28] used sparse Bayesian learning. Neural networks have been proposed in [22, 24, 29, 30], while Trajectory planning methods of lane changes for urban scenarios can be found in [31, 32]. Recently, an alternative approach to the labeling procedure in the creation of the dataset has been investigated [33, 34]. Rákos et al. [35] implemented a mixed procedure with Gaussian Classification, Support Vector Classification and Neural Network Classifiers. Finally, Chen et al. [36] used a technique known as LightGBM.

Among all the techniques available in the literature and briefly mentioned in this section, we decided to use SVM [37] and Random Forests [38]. We chose SVM for a number of reasons. First, as pointed out by Mandalia [21] and Li [23], they are suited to address problems with temporal series and data coming from each driving test form a temporal series. Second, some computational experiments proved that SVM perform better than Hidden Markov Models [18]. Finally, SVM offer the great advantage to map the original feature set over a higher-dimensional space. This is very effective in terms of separability. This last point perfectly suits with the preprocessing techniques described in Sect. 3. In particular, the SMOTETomek algorithm [14, 15] outlined in Sect. 3 fosters separability by deleting those instances in the dataset which would fall within a region with a different label. Instead, Random Forests are trained by creating rules based on the concept of information gain indicating the usefulness of each feature. Both approaches fit very well with the SMOTETomek preprocessing we use, favoring greater separability. Therefore, we believe that the use of the SMOTETomek technique in concert with the proposed classifiers is a good combination for efficient predictions.

3 Case Study

In this section, we show the development of our decision model in concert with a case study. The fundamental idea is to train classifiers able to predict the next maneuver online through the classification of data coming from sensors the machine is equipped with. In order to accomplish this decision, it is important to train the classifiers on a dataset that is representative of real driving conditions. During the European project DESERVE such a dataset was collected thanks to the highly-technological demonstrator vehicle provided by CRF (see Fig. 1).

This vehicle is able to provide a number of features through sensors at a sampling time of 50 ms. In particular, the installed technology consists of: internal camera able to measure head position and eye movements of the driver, external camera able to capture the road and radar able to detect the next vehicle. Other features are: lateral distance, curvature, brake pedal, gas pedal, yaw rate, vehicle speed, lateral distance, steering angle, heading angle and engine speed. 43 volunteers drove for about 1 h and a half on the demonstrator vehicle on the Italian A55 Torino-Pinerolo highway (Fig. 1). Every driver performed around 20 overtaking maneuvers for a total of more than 800 overtaking maneuvers. Raw data coming from sensors produce, however, noisy measurements. Therefore, it cannot be used as immediate input for training a classifier and so a preprocessing step is needed. Preprocessing is a procedure used to reduce noise from measurements and prepare data for the subsequent training procedure. A common solution is to compute a sequence of feature statistics on sliding windows. Inspired by the approach used by Mandalia and Salvucci [21], we used shifting time windows. In this approach, data from each driver is considered as a time series. For each time series, a shifting time window is used to compute the variance among its samples in order to address two drawbacks regarding raw data. First of all it is useful to reduce sensors' noise. Second, it is helpful to manage fluctuations in data. In fact, only in an ideal situation a driver

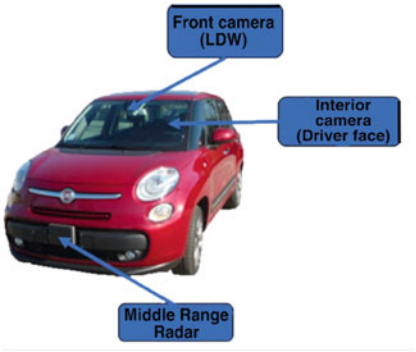


Fig. 1 The demonstrator vehicle provided by Centro Ricerche Fiat (left) and a section of the Italia A55 highway where driving tests took place (right)

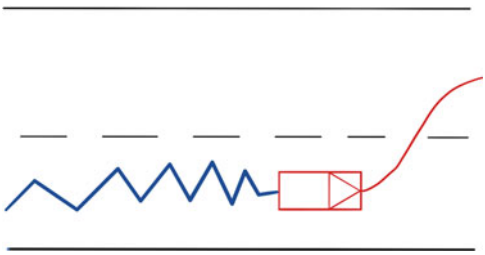


Fig. 2 Data swinging typical of real driving

perfectly drives in straight line. In reality, driving is characterized by some small oscillations (see Fig. 2). This phenomenon can be considerably reduced if variance of features is used instead of raw data.

As we have already pointed out, the overall dataset is strongly unbalanced. This is because a driver is mainly in the so-called lane keeping state, while he or she rarely performs a lane change maneuver. Therefore, the cardinality of the instances labeled as lane change is significantly lower than the cardinality of the instances labeled as lane keeping. To solve this problem, we used the SMOTETomek algorithm [14, 15]. This algorithm combines oversampling and undersampling techniques. The former addresses the problem of unbalanced datasets while the latter fosters separability removing unwanted overlap between classes. This increases the space between two classes, helping the classification process.

Preprocessing is also a procedure useful to detect the most significant features. At this stage of our case study, we have decided not to consider data from cameras in order to answer the question of whether it is possible to train an efficient classifier with the data available from most vehicles. At the moment, in fact, most vehicles do not have an internal or external camera. We therefore aim to train a decision-making model that improves safety for all vehicles, not just the most equipped ones. In the preprocessing phase, we also conducted an analysis of the most significant features, which turned out to be: heading angle, curvature, vehicle speed, yaw rate, steering angle and engine speed. This subset of features is in line with other works in the literature. In particular, lateral distance and heading angle are two fundamental features to predict a lane change. However, the cameras were a fundamental control tool for the creation of the dataset.

4 Computational Results

In this section, we report the results of the computational tests performed on our classifiers. These tests were performed on a workstation with a 2.3 GHz processor and a 8 GB RAM using the sklearn package [39] of the Python programming language. Training, validation and testing sets have been set respectively considering 50%, 20% and 30% of drivings. This partition corresponds to 990,930 instances for the training set, 501,160 for the validation set and 464,257 for the test set, for a total

Table 1 Performances of the classifiers used

Labels	Metrics	SVM	Random forest
2	Gini	72%	79%
	Precision	82%	89.9%
	Recall	81.3%	89.5%
	F1-measure	82%	89.7%
	Accuracy	83%	89.7%
3	Gini	72%	80%
	Precision (micro)	82.1%	87.1%
	Precision (macro)	81.8%	87.8%
	Recall (micro)	81.5%	87.3%
	Recall (macro)	82.1%	87.1%
	F1-measure (micro)	81.4%	87.4%
	F1-measure (macro)	81.6%	87.2%
	Accuracy	83%	87.2%

of 1,956,347 instances. We used the validation set to calibrate the parameters of the classifiers. For SVM, these are the kind of kernel, the penalty parameter and the gamma parameter of the radial basis function (RBF) kernel. For Random Forests, these are the number of decision trees, the maximum depth and the criterion used (entropy or Gini). We used a grid search procedure to tune these parameters. The best combination of parameters for SVM turned out to be a RBF kernel with penalty parameter equal to 100 and gamma coefficient equal to 1. For Random Forests, we found that the best combination of parameters is a Gini criterion with 100 decision trees and a maximum depth of 30.

Table 1 shows the performance of the classifiers used with both two and three labels.

We wish to point out that for the multiclass case we used both micro and macro metrics. With two labels, precision is defined as the portion of positive and correct identifications, i.e. PREC $= \frac{TP}{TP+FP}$, where TP is the number of true positives and FP is the number of false positives. With two labels, one class is assumed to be positive and the other one is assumed to be negative. When more than two labels are present, one label can in turn be considered as positive and the remaining ones as negative. Let L denote the number of labels. In this way, one can compute $TP(l)$ and $FP(l)$ which respectively are the number of true positives and false positives when label $l \in L$ is assumed as positive and the remaining labels as negative. Micro precision is defined as

$$\text{PREC (micro)} = \frac{TP_{\text{sum}}}{TP_{\text{sum}} + FP_{\text{sum}}} = \frac{\sum_{l \in L} TP(l)}{\sum_{l \in L} TP(l) + \sum_{l \in L} FP(l)}, \tag{1}$$

Table 2 Prediction time for the proposed classifiers

Labels	Maneuver	SVM	Random forest
2	Lane change	1.07 s	2.19 s
3	Lane change left	0.88 s	1.15 s
	Lane change right	0.95 s	1.57 s

while macro precision is computed as PREC (micro) $= \frac{1}{L}\sum_{l \in L} \text{PREC}(l)$, where PREC($l$) is the precision when label $l \in L$ is assumed as positive label, i.e., PREC(l) $= \frac{TP(l)}{TP(l)+FP(l)}$. Similar considerations can be done for recall and F1-measure [40]. We can see that both classifiers show very good performances. SVM shows an accuracy of 83% both in the case with two and three labels, while the accuracy of the random forest slightly decreases from 89.7% to 87.2%. From these results we can also infer that Random Forest performances are higher than SVM ones. Nevertheless, SVM performances practically remain the same if a third label is added, whilst Random Forest performances tend to decrease if the discrimination between left and right lane change is introduced.

In addition to the classic figures of merit shown in Table 1, we also computed the prediction time for a lane change. This is defined as the time before a lane change during which a classifier succeeds in predicting the lane change maneuver. The results are shown in Table 2.

Again, we can observe that Random Forest shows better performances than SVM. However, the latter is more stable if three labels are introduced. Moreover, we can observe an interesting pattern: on average the prediction time for the right lane change is higher than the prediction time for the left lane change. This behavior perfectly reflects what happens in reality, where, in general, left lane change maneuvers are faster than right lane change maneuvers. In fact, left lane changes tend to be more rapid because the driver has to look at the vehicles in the next lane. Vice versa, right lane changes are a return maneuver and tend to be calmer.

Finally, in Tables 3 and 4, we propose a comparison with some works in the literature. It is our concern to point out that a true comparison is technically not possible because the datasets considered are different. In fact, some data come from simulation, others from real roads, others with pieces of highways, and so a concrete comparison is not really possible. Thus, the aim of these tables is to prove that our results are in line with other works in the literature. Table 3 shows the case with two labels (i.e., lane keeping and lane change), while Table 4 refers to the case with three labels, namely lane keeping and left and right lane change. Please also note that, concerning our results for the case with three labels, we reported the average prediction time between left and right lane change, which is 0.915s for SVM and 1.36s for Random Forest. The literature on the subject is copious and therefore we have just selected some of the most popular works.

Table 3 Comparisons with other works in the literature using two labels

Author	Method	Kind of data	Best accuracy	Prediction time
Dagli et al. [20]	Bayesian Network	Simulator	80%	1.5 s
Salvucci [26]	Mind-tracing	Realistic highway simulation	82%	1.1 s
Mandalia and Salvucci [21]	SVM	Real data	87%	0.3 s after lane change starts
Dogan et al. [22]	Neural network	NISYS TRS	unknown	1.5 s
Shou et al. [30]	RNN	NGSIM	75%.	8.05 s
Our work	SVM	Real highway data	83%	1.07 s
	Random forest	Real highway data	89.7%	2.19 s

Table 4 Comparisons with other works in the literature using three labels

Author	Method	Kind of data	Best accuracy	Prediction time
Li [23]	SVM	Simulator	63.3%	1.0447 s
Augustin et al. [33]	Boosted decision tree	Real-world	94%	1.67 s
Benterki et al. [24]	ANN	NGSIM	97.1%	2.33 s
Rákos et al. [35]	NN and SVM	NGSIM	86%	Unknown
Our work	SVM	Real highway data	83%	0.915 s
	Random forest	Real highway data	87.2%	1.36 s

5 Conclusions

In this paper, we have introduced a decision model to improve safety while driving. The main contributions of this work concern the real dataset consisting of 43 real driving tests on highways and having carried out a study with both two and three labels. The latter case allowed us to distinguish between left and right lane changes: an important aspect for highway driving. The decision model is rooted in the DESERVE European project to improve road safety through embedded Advanced Driver Assistance Systems. This article is intended as a starting point for future work. One possible development may be to consider features from internal and external cameras or other labels of interest related to lane change maneuvers such as complete overtakings or car followings. Another possible development planned is the use of ensemble learning techniques rather than independent classifiers. We are confident that the use of these techniques will lead to further improvements of the proposed results, which are already very satisfactory and in line with those present in the literature.

References

1. Eurostat Statistics Explained: Passenger transport statistics. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=passenger_transport_statistics
2. Eurostat Statistics Explained: Passenger transport statistics. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=transport
3. Lefèvre, S., Vasquez, D., Laugier, C.: A survey on motion prediction and risk assessment for intelligent vehicles. Robomech 1 (2014). https://doi.org/10.1186/s40648-014-0001-z
4. Baldi, M.M., Perboli, G., Tadei, R.: Driver maneuvers inference through machine learning. In: Machine Learning, Optimization, and Big Data, pp. 182–192. Springer, Berlin (2016)
5. The European Deserve project: https://www.deserve-project.eu/
6. Centro Ricerche Fiat: https://www.crf.it/en.
7. He, Z.: Research based on high-fidelity NGSIM vehicle trajectory datasets: a review (2017)
8. Thiemann, C., Treiber, M., Kesting, A.: Estimating acceleration and lane-changing dynamics from next generation simulation trajectory data. Transp. Res. Rec. **2088**(1), 90–101 (2008)
9. Hamdar, S., Mahmassani, H.: Driver car-following behavior: From discrete event process to continuous set of episodes. In: Proceedings of the 87th Annual Meeting of the Transportation Research Board (2008)
10. Duret, A., Buisson, C., Chiabaut, N.: Estimating individual speed-spacing relationship and assessing ability of Newell's car-following model to reproduce trajectories. Transp. Res. Rec. **2088**(1), 188–197 (2008)
11. Punzo, V., Borzacchiello, M.T., Ciuffo, B.: On the assessment of vehicle trajectory data accuracy and application to the next generation simulation (NGSIM) program data. Transp. Res. C: Emer. Technol. **19**(6), 1243–1262 (2011)
12. Montanino, M., Punzo, V.: Trajectory data reconstruction and simulation-based validation against macroscopic traffic patterns. Transp. Res. B: Methodol. **80**, 82–106 (2015)
13. Coifman, B., Li, L.: A critical evaluation of the next generation simulation (NGSIM) vehicle trajectory dataset. Transp. Res. B: Methodol. **105**, 362–377 (2017)
14. Batista, G.E.A.P.A., Bazzan, A.L.C., Monard, M.A.: Balancing training data for automated annotation of keywords: case study. In: Proceeding of the Second Brazilian Workshop on Bioinformatics, pp. 35–43 (2003)
15. Imbalanced-learn documentation. https://imbalanced-learn.org/stable/
16. Liu, A., Pentland, A.: Towards real-time recognition of driver intentions. In: Proceedings of Conference on Intelligent Transportation Systems, pp. 236–241 (1997)
17. Kuge, N., Yamamura, T., Shimoyama, O., Liu, A.: A driver behavior recognition method based on a driver model framework. In: SAE Technical Paper Series, pp. 2000-01-0349 (2000)
18. Mandalia, H.M.: Pattern recognition techniques to infer driver intentions. Ph.D. Thesis, Drexel University, 2004
19. Oliver, N., Pentland, A.P.: Driver behavior recognition and prediction in a smartcar. In: Verly, J.G. (ed.) Enhanced and Synthetic Vision 2000, vol. 4023, pp. 280–290. International Society for Optics and Photonics, SPIE (2000)
20. Dagli, I., Brost, M., Breuel, G.: Action recognition and prediction for driver assistance systems using dynamic belief networks. In: Carbonell, J.G., Siekmann, J., Kowalczyk, R., Müller, J.P., Tianfield, H., Unland, R. (ed.) Agent Technologies, Infrastructures, Tools, and Applications for E-Services, pp. 179–194. Springer, Berlin (2003)
21. Mandalia, H.M., Salvucci, D.: Using support vector machines for lane-change detection. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 49, pp. 1965–1969. SAGE Publications, Los Angeles (2005)
22. Dogan, U., Edelbrunner, H., Iossifidis, I.: Towards a driver model: Preliminary study of lane change behavior. In: 2008 11th International IEEE Conference on Intelligent Transportation Systems, pp. 931–937. IEEE, Piscataway (2008)
23. Li, Z.: Prediction of vehicles' trajectories based on driver behavior models. Master's Thesis, Delft University of Technology, 2014

24. Benterki, A., Boukhnifer, M., Judalet, V., Choubeila, M.: Prediction of surrounding vehicles lane change intention using machine learning. In: 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), vol. 2, pp. 839–843 (2019)
25. Li, M., Li, Z., Xu, C., Liu, T.: Short-term prediction of safety and operation impacts of lane changes in oscillations with empirical vehicle trajectories. Accid. Anal. Prevent. **135**, 105345 (2020)
26. Salvucci, D.: Inferring driver intent: A case study in lane-change detection. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 48, pp. 2228–2231. SAGE Publications, Los Angeles (2004)
27. Salvucci, D., Mandalia, H.M., Kuge, N., Yamamura, T.: Lane-change detection using a computational driver model. Hum. Fact. **49**(3), 532–542 (2007)
28. McCall, J.C., Wipf, D.P., Trivedi, M.M., Rao, B.D.: Lane change intent analysis using robust operators and sparse Bayesian learning. IEEE Trans. Intell. Transp. Syst. **8**(3), 431–440 (2007)
29. Han, T., Jing, J., Özgüner, Ü.: Driving intention recognition and lane change prediction on the highway. In: 2019 IEEE Intelligent Vehicles Symposium (IV), pp. 957–962. IEEE, Piscataway (2019)
30. Shou, Z., Wang, Z., Han, K., Liu, Y., Tiwari, P., Di, X.: Long-term prediction of lane change maneuver through a multilayer perceptron. In: 2020 IEEE Intelligent Vehicles Symposium (IV), pp. 246–252. IEEE, Piscataway (2020)
31. Li, A., Li, S., Shen, H., Huan, M., Miao, X., Li, X.: Motion trajectory planning method of lane change for intelligent vehicle. J. Theor. Appl. Inform. Technol. **45**(1), 297–302 (2012)
32. Chandru, R., Selvaraj, Y.: Motion planning for autonomous lane change manoeuvre with abort ability. Master's Thesis, Chalmers University of Technology, 2016
33. Augustin, D., Hofmann, M., Konigorski, U.: Motion pattern recognition for maneuver detection and trajectory prediction on highways. In: 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES), pp. 1–8 (2018)
34. Augustin, D., Hofmann, M., Konigorski, U.: Prediction of highway lane changes based on prototype trajectories. Forsch. Ingenieurwes. **83**(2), 149–161 (2019)
35. Rákos, O., Aradi, S., Bécsi, T.: Lane change prediction using Gaussian classification, support vector classification and neural network classifiers. Period. Polytech. Transp. Eng. **48**(4), 327–333 (2020)
36. Chen, T., Shi, X., Wong, Y.D., Yu, X.: Predicting lane-changing risk level based on vehicles' space-series features: a pre-emptive learning approach. Transp. Res. C: Emer. Technol. **116**, 102646 (2020)
37. Boser, E.B., Guyon, I.M., Vapnik, N.V.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92, pp. 144–152. Association for Computing Machinery (1992)
38. Ho, T.K.: Random decision forests. In: Proceedings of 3rd International Conference on Document Analysis and Recognition, vol. 1, pp. 278–282 (1995)
39. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O. Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
40. Murphy, K.P.: Machine Learning, a Probabilistic Perspective. MIT Press, Cambridge (2012)

A Two-Echelon Truck-and-Drone Distribution System: Formulation and Heuristic Approach

M. Boccia, A. Mancuso, A. Masone, A. Sforza, and C. Sterle

Abstract The integration of new distribution technologies in the delivery systems, specifically drones, has been investigated by several companies to reduce the Last-Mile Logistics costs. In this context, different truck-and-drone delivery systems have been proposed in literature, where the truck operates as a mobile depot for the drones. In this work, we study a two-echelon truck-and-drone distribution system where the first-echelon is composed of the depot and truck parking places, whereas the second-echelon is composed of the parking places and the final customers that are served by a fleet of drones. The truck parking places can be considered as locations where the operator can deploy the fleet of drones and monitor their trips. Moreover, we propose a mixed integer linear programming formulation and a two-stage heuristic that exploits the underlying structure of the problem. The proposed approaches are tested and validate on set of instances up to 50 customers.

Keywords Unmanned aerial vehicle · Location routing problem · Drone-truck combined operations · Transportation planning

1 Introduction

Last mile logistics (*LML*), is probably the most inefficient and expensive phase of a distribution process, because of the dynamic nature of the urban environment and economic activities [9]. In [6], the major challenges, trends and innovative delivery systems for *LML* are discussed, considering different aspects such as technological and infrastructural issues, system design and management, and logistic costs. In this context, the integration of Unmanned Aerial Vehicles (UAVs) in the *LML* delivery systems received great attention by the scientific research community [16].

M. Boccia · A. Mancuso · A. Masone (✉) · A. Sforza · C. Sterle
Department of Electrical Engineering and Information Technology, University of Naples "Federico II", Naples, Italy
e-mail: maurizio.boccia@unina.it; andrea.mancuso@unina.it; adriano.masone@unina.it; sforza@unina.it; claudio.stelre@unina.it

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_13

The motivations behind this growing research interest can be mainly found in: the continuous developments of drone technologies; the exploration of new drone applications driven by large corporations such as Amazon, FedEx, and UPS [7]; the achievable sustainability targets (e.g. emissions and delivery completion time reduction) which benefit of drone usage in *LML* [14]. This is also coherent with the Green Transition promoted by the European Commission in the forthcoming Horizon Europe 2021–2027 programme.

The most studied drone assisted logistic system consists of a single truck and a single drone operating in tandem for the parcel delivery to the customers [1, 4, 5]. In a nutshell, it performs as follows: a drone departs from a truck, delivers a parcel, and returns to the truck without any human intervention; the truck can serve customers along its route and operates as a mobile hub for the drone. The main benefit of this system consists in the completion time reduction achievable through the vehicles synchronization and workload parallelization.

However, current safety and aerial national regulations of several countries impose restrictions on UAV civil usage in terms of weather conditions and associated risks (e.g., collision and crashing, deliberate attack, payload theft). In this context, one of the most common restriction is either that drones should fly in the line of sight of a human operator or they could fly only within limited service area [6].

To meet these requirements, different truck-and-drone delivery systems have been proposed in literature, where the truck operates as a mobile depot [8, 13, 18]. More precisely, the truck stops at one or more location (referred to as station in the following), where one or more drones are launched to perform the delivery last leg. The truck performs no delivery operation and it has to collect all the launched drones before moving from a location to the successive one. Thus, the synchronization issue between the two kinds of vehicles can be neglected. In [8], the stations to be visited by the truck have to be determined in a continuous space. Hence, the truck is not constrained by the road network. The objective is to find the distribution plan minimizing the completion time. The resulting optimization problem is solved by a cluster first-route second heuristic, refined by a local search. A similar problem considering a single drone is tackled in [13]. The authors develop a two stage heuristic method which determines the stations by K-means algorithm and solves the deriving traveling salesman problem (*TSP*) by a genetic algorithm. Finally, in [18], the authors extend the problem studied in [8] considering the possibility that a drone can make more than one trip. They present a Mixed Integer Non-Linear Programming formulation and study the effect of different side constraints on the achieved solutions.

In this work, we study a variant of the problem addressed in [8], where the truck is constrained to move on a road network whose nodes represent potential stations to be used. As explained above, these nodes represent truck parking places where the operator can deploy the fleet of drones and monitor their trips ideally without losing the line-of-sight. We propose a mixed integer linear programming (*MILP*) formulation and a two stage heuristic method exploiting the underlying structure of

the problem. The proposed approaches are tested and validate on a set of instances up to 50 customers.

For the sake of the completeness, we highlight that the tackled problem falls in the class of two-echelon location-routing problem (*2E-LRP*), where the first-echelon is composed by the depot and stations, whereas the second-echelon is composed by the stations and the final customers. Using the *LRP* notation proposed in [2], it can be classified as a $3/\overline{T}/R$ problem, i.e., a two-echelon location-routing problem, with location decisions on the second layer, multiple node routes on the first-echelon and direct routes on the second-echelon. In the following we refer to the tackled problem as two-echelon truck-and-drone location-routing problem (*2E-TD-LRP*). Most of the *2E-LRP* literature deals with $3/\overline{R}/T$, $3/\overline{T}/T$, $3/\overline{R}/\overline{T}$ and $3/\overline{T}/\overline{T}$ and the objective is the minimization of the overall transportation cost. Unlike, in the tackled problem the aim is determining the distribution plan minimizing the completion time of the delivery process. For this reason and for the sake of the brevity we do not provide a review of *2E-LRP* literature but we just address the reader to [11] and [3].

The paper is organized as follows: in Sect. 2, a description of the *2E-TD-LRP* is provided; Sect. 3 is devoted to the presentation of the *MILP* formulation; Sect. 4 presents the proposed heuristic method; Sect. 5 shows the obtained computational results; finally, conclusions and future perspectives are given in Sect. 6.

2 Problem Description

The *2E-TD-LRP* is based on the following delivery conditions and operating assumptions:

- The truck departs and returns to a single depot exactly once.
- Each station can be visited at most once by the truck.
- The truck has an infinite capacity and acts as a mobile depot for the fleet of drones.
- Each customer must be served only once by a drone.
- Each drone can serve one customer per sortie. Thus, the number of customers that can be served from a station is limited by the drone fleet size.
- A drone can be launched only at the stations when the truck is stationary.
- A drone has to be launched and collected at the same station.
- Drone flight time is limited because of the battery power (endurance).
- Each drone is supervised by the truck operator during the whole sortie.
- The truck can continue its route only if the whole fleet is on board.
- The travel time of the drones and the truck are different because of the different speeds.

On this basis, the *2E-TD-LRP* entails three kinds of decisions:

- location decisions: selecting the stations to be used by the truck for parking;
- assignment decisions: determining the assignment of the customers to the selected stations;
- routing decisions: definition of the truck tour visiting the selected stations.

The aim is the minimization of the delivery completion time that can be expressed as the sum of truck traveling time and the maximum drone flight time in each node station. Indeed, since the truck can continue its route only when all drones are back on board and the fleet of drones serve customers in parallel, then the truck has to wait until all the deployed drones have completed their deliveries. Therefore, in other words, the truck waiting time in each station is given by the maximum delivery time among all the drones serving a customer from the selected stations.

3 Problem Formulation

To introduce the *MILP* formulation for the *2E-TD-LRP*, let us consider a symmetric network $G(V, A)$, where: $V = \{o \cup N \cup C\}$ is the set of nodes composed by the depot o (where the truck is initially located), the set N of station nodes and the set C of the customers; A is the set of the arcs composing the network.

A cost t_{ij}^T and a cost t_{ij}^D are associated to each couple of nodes $(i, j), \forall i, j \in N$ and $(i, k), \forall i \in N, j \in C$, respectively. These costs indicate the time required to move from node i to node j either by the truck or by the drone. The drone endurance is expressed in terms of maximum travelling time indicated by the parameter t_{max}. Moreover, let $D = \{1, .., m\}$ be the set of drones available on the truck.

On this basis, we introduce the following decision variables:

- $y_j, \forall j \in N$, binary variable equal to 1 if the station node j is used for the drone deliveries, 0 otherwise (location variable);
- $x_{ij}, \forall i, j \in N$, binary variable equal to 1 if the truck travels from the station node i to station node j, 0 otherwise (routing variable);
- $z_{ij}^d, \forall i \in N \setminus \{o\}, j \in C, d \in D$, binary variable equal to 1 if customer j is served by drone d from station node i, 0 otherwise (assignment variable);
- $w_i, \forall i \in N \setminus \{o\}$, continuous variable equal to the truck waiting time for the drone deliveries starting from the station node i.

Then, the *2E-TD-LRP* can be modeled as follows:

$$Min \sum_{i,j \in N} t_{ij}^T x_{ij} + \sum_{i \in N \setminus \{o\}} w_i \tag{1}$$

$$s.t.$$

$$z_{jk}^d \leq y_j \qquad \forall j \in N \setminus \{o\}, k \in C, d \in D \tag{2}$$

$$2t_{jk}^{D} z_{jk}^{d} \leq t_{max} \qquad \forall j \in N \setminus \{o\}, k \in C, d \in D \tag{3}$$

$$\sum_{d \in D} \sum_{k \in C} z_{jk}^{d} \leq m \qquad \forall j \in N \setminus \{o\} \tag{4}$$

$$\sum_{j \in N \setminus \{o\}} \sum_{d \in D} z_{jk}^{d} = 1 \qquad \forall k \in C \tag{5}$$

$$\sum_{k \in C} z_{jk}^{d} \leq 1 \qquad \forall j \in N \setminus \{o\}, d \in D \tag{6}$$

$$w_j \geq 2t_{jk}^{D} z_{jk}^{d} \qquad \forall j \in N \setminus \{o\}, k \in C, d \in D \tag{7}$$

$$\sum_{i \in N | i \neq j} x_{ij} = y_j \qquad \forall j \in N \tag{8}$$

$$\sum_{j \in N | j \neq i} x_{ij} = \sum_{l \in N | l \neq l} x_{jl} \qquad \forall j \in N \tag{9}$$

$$\sum_{i \in S} \sum_{j \in N \setminus S} x_{ij} = y_u + y_v - 1 \qquad \forall S \subseteq N, u \in S, v \in N \setminus S \tag{10}$$

$$y_o = 1 \tag{11}$$

$$x_{ij}, y_j \in \{0, 1\} \qquad \forall i, j \in N \tag{12}$$

$$z_{jk}^{d} \in \{0, 1\} \qquad \forall j \in N \setminus \{o\}, k \in C, d \in D \tag{13}$$

$$w_i \geq 0 \qquad \forall i \in N \setminus \{o\} \tag{14}$$

The objective function (1) minimizes the delivery completion time expressed as sum of the truck route duration and the truck waiting times in each station node selected for the drone deliveries. Constraints (2) are consistency constraints linking location and assignment variables. Constraints (3) impose a limit on the flying time of the drones with respect to their endurance. Constraints (4) impose that the maximum number of drones to be launched from a station node cannot exceed the number of available nodes. Constraints (5) ensure that each customer has to be served exactly once. Constraints (6) impose that each drone can be launched only once from each station. Constraints (7) set the truck waiting time in each station node. Constraints (8) guarantee that the truck route will visit all the selected station nodes. The feasibility of the truck route in terms of flow conservation and subtour elimination is ensured by constraints (9) and (10), respectively. Finally, constraints (12)–(14) express the nature of the variables.

4 A Two-Stage Solution Approach

It is widely known that *LRP*s belong to the class of *NP*-Hard problems as reported in [11, 12]. Indeed, as expected, the proposed *2E-TD-LRP* formulation can be hard to solve using a *MILP* solver even for small instances. Therefore, we develop a solution method that exploits the underlying structure of the problem, which can be naturally decomposed in two sub-problems dealing with location and assignment decisions and routing decisions, respectively. In particular, the first sub-problem consists in the determination of the station nodes to be used and the resulting customer assignment which satisfies drone fleet size constraints and minimizes drone delivery time. The second sub-problem consists in the determination of the minimum length truck tour visiting all the selected station nodes.

On the basis, we developed a two-stage solution approach that in the first-stage selects the station nodes and determines the customer assignment through the solution of a facility location problem (*FLP*) [17]. Then, the *FLP* solution is used as input for the second-stage, where a *TSP* between the selected station nodes is solved. Finally, the obtained solution is further optimized by a local search procedure which modifies the selected station nodes at first-stage. The details of the proposed method are discussed in the following.

First Stage: Station Node Selection and Customer Assignment
In *FLP*s, there are a set of potential locations for facilities with fixed costs and capacities, and a set of customers with known demands of the goods to be supplied. The unit transportation cost for goods supplied from the facilities to all the customers is given. The aim is to determine the subset of facilities minimizing the total transportation cost such that demand of all the customers can be satisfied without violating the capacity constraints. Three of the most studied *FLP*s are: the p-Center Problem (p-*CP*, [10]), the p-Median Problem (p-*MP*, [15]), and Simple Plant Location Problem (*SPLP*, [19]). The first two problems impose that the number of facilities to be located has to be equal to a pre-fixed value p. The p-*CP* minimizes the maximum transportation cost between the customers and the facilities, while the p-*MP* minimizes the sum of the overall transportation costs from facilities to customers. On the other hand, the *SPLP* does not fix the number of plants and minimizes the sum of location and transportation costs.

In our problem, the station nodes can be conceived as facilities. Each station node has a capacity equal to the number of drones available on the truck (m), while all the customers demands are equal to 1. The transportation cost from the facilities to all the customers corresponds to the drone travelling time from the station nodes to the customers. Moreover, unlike the *FLP*s, we have additional constraints related to the drone endurance which prevent customer-to-station assignments out of the drone operation range.

On this basis, it is clear that our sub-problem slightly differs from any *FLP* known in literature. Therefore, we considered three of the most studied *FLP*s with the aim of identifying the most suitable one for our problem. To this aim, we developed three versions of the proposed heuristic differing in the solved *FLP* in the first stage.

Concerning the *p-CP* and the *p-MP*, since we do not know in advance the number of station nodes to be selected, we defined a procedure to determine the p value. Being C_j the subset of customers reachable by a drone launched from node station j, the procedure may be schematized as follows:

1. Compute t_j^* as the average drone transportation cost from node j to all the reachable customers ($t_j^* = \sum_{k \in C_j} t_{ij}^D / |C_j|$).
2. Compute t^* as the average drone transportation cost ($t^* = \sum_{j \in N} t_j^* / |N|$).
3. Define p as the number of node stations j such that $t_j^* \leq t^*$.

Then, based on the notation defined in Sect. 3, the objective function considered for the *p-CP* is:

$$Min_{i \in N \setminus \{o\}} w_i \tag{15}$$

while for the *p-MP* is:

$$Min \sum_{i \in N \setminus \{o\}, j \in C_i, d \in D} 2 t_{ij} z_{ij}^d \tag{16}$$

Concerning instead the *SPLP*, since we do not have a proper location cost for the station nodes, we defined it as follows:

$$h_j = \sum_{i \in N} t_{ij}^T / |N| - 1, \forall j \in N \setminus \{o\} \tag{17}$$

Therefore, we can express the *SPLP* objective function as follows:

$$Min \sum_{i \in N \setminus \{o\}} h_i y_i + \sum_{i \in N \setminus \{o\}, j \in C_i, d \in D} 2 t_{ij} z_{ij}^d \tag{18}$$

Second Stage: Truck Route Determination

The *TSP* consists in determining the minimum length tour connecting a set of n cities. As explained above, in the *2E-TD-LRP* the truck has to perform a tour connecting the selected station nodes minimizing the travelling time. Thus, in the second-stage of the proposed method, we solve a *TSP* over the set $O \bigcup \{o\}$, where O is the set of station nodes determined by the *FLP* solution in the firs-stage and $\{o\}$ is the depot.

Local Search for the Node Station Location

Preliminary results showed that on some instances, the number of station nodes selected in the optimal solutions and the one determined in the first-stage of the heuristic differs of few units (usually 1 or 2). Therefore, we developed a simple local search (*LS*) that at each iteration increases by 1 the number of station nodes in the heuristic solution. In particular, at the first iteration, being p^* the number of station nodes determined in the initial *FLP* solution, the *LS* solves the same *FLP*

fixing the number of station nodes to $p^* + 1$. Then, the truck route among the new p^*+1 station nodes is built solving a *TSP*. The *LS* continues until it finds a solutions with a better objective function value.

5 Computational Results

The proposed approaches have been implemented in Python 3.6 and linked with *MILP* solver (FICO-Xpress optimizer 8.9). The *FLP* formulations are solved by the *MILP* solver. On the other hand, we point out that we developed a branch-and-cut algorithm to solve the proposed *MILP* formulation and the *TSP* formulations since they present subtour elimination constraints that are exponential in number. The computational experiments were carried out with a time limit of 3600 s on an Intel Core I7-8700 CPU 3.20 GHz workstation with 16 Gb of RAM.

For our test bed, we generated 40 instances. In particular, we considered two different size for the station node set ($|N| = 10, 20$). For each value of $|N|$, we generated 10 instances with 30 customers and 5 instances with 40 and 50 customers, respectively. The depot, the station nodes and the customers are randomly distributed across an 8-mile square area. The speed of the truck and of the drones are equal to 25 miles/h and 35 miles/h, respectively. Further, we considered the truck equipped with 10 drones on-board, each one with an endurance equal to 20 min.

In our experimentation, we tested the branch-and-cut (*B&C*) algorithm developed to solve the *MILP* formulation and the three versions of the proposed two-stage solution method obtained considering in the first stage the *p-CP*, *p-MP*, and *SPLP*, respectively.

Tables 1 and 2 report the results of the *B&C* and the best results among the three versions of the heuristic method (*FLP+TSP+LS*) on the instances with $|N| = 10$ and $|N| = 20$, respectively. For each instance, the tables report in the first column the instance id (*ID*). The following columns report for both approaches: the delivery completion time in minutes (*Ct*), the corresponding percentage of truck (*T%*) and drone (*D%*) travelling time, the solution running time in seconds (*Rt*). The percentage of truck (drone) travelling time is computed as $T\%\ (D\%) = truck\ (drone)\ time\ /\ Ct \cdot 100$, where the truck and drone times are the first and the second component of the *MILP* objective function, respectively.

All the instances have been solved to optimality within the imposed time limit.

In terms of delivery system performance, we can observe that on average the truck and drone traveling times are approximately the 40% and 60% of the completion time, respectively. This workload distribution confirms the benefit arising from this hybrid delivery system in terms of emissions being drones less polluting than trucks.

Table 1 Results of the proposed approaches on the instances with $|N| = 10$

		B&C				FLP+TSP+LS				
ID	$\|C\|$	Ct	$T\%$	$D\%$	Rt	Ct	$T\%$	$D\%$	Rt	$Gap\%$
1	30	86	50.00	50.00	336	86	48.84	51.16	28.4	0.00
2	30	83	40.96	59.04	891	83	40.96	59.04	407	0.00
3	30	92	47.83	52.17	803	93	38.71	61.29	312	1.09
4	30	91	51.65	48.35	184	91	51.65	48.35	167	0.00
5	30	72	34.72	65.28	219	76	47.37	52.63	182	5.56
6	30	97	44.33	55.67	459	97	44.33	55.67	187	0.00
7	30	91	32.97	67.03	231	91	32.97	67.03	162	0.00
8	30	57	33.33	66.67	489	57	33.33	66.67	123	0.00
9	30	56	37.50	62.50	265	56	37.50	62.50	28	0.00
10	30	67	44.78	55.22	181	67	44.78	55.22	132	0.00
11	40	94	46.81	53.19	1431	94	46.81	53.19	155	0.00
12	40	73	41.10	58.90	148	73	41.10	58.90	129	0.00
13	40	63	39.68	60.32	196	63	39.68	60.32	181	0.00
14	40	91	38.46	61.54	308	94	45.74	54.26	210	3.30
15	40	91	36.26	63.74	441	91	36.26	63.74	201	0.00
16	50	91	41.76	58.24	409	91	41.76	58.24	190	0.00
17	50	70	37.14	62.86	698	74	48.65	51.35	256	5.71
18	50	71	53.52	46.48	274	71	53.52	46.48	182	0.00
19	50	77	41.56	58.44	361	79	49.37	50.63	250	2.60
20	50	88	53.41	46.59	251	88	53.41	46.59	136	0.00
Mean		80.05	42.39	57.61	428.75	80.75	43.84	56.16	180.92	0.91

In terms of solution approaches, we can observe from Table 1 that the proposed heuristic is able to optimally solve 15 out of 20 instances with an average percentage gap equal to 0.91. Moreover, the total running time of the heuristic is half the running time of the *B&C* algorithm, even if it requires the solution of two *MILP* formulations multiple times. The effectiveness of the heuristic method is further confirmed by the results of Table 2. Indeed, the proposed heuristic approach is able to solve 18 out of the 20 instances with $|N| = 20$ resulting in an average percentage gap of 0.25. Moreover, the heuristic running times are one third of the *B&C* algorithm proving a good scalability of the heuristic approach in terms of computation of times. In particular, we observed that the solution of the *FLP* represents, on average, the 30% of the heuristic running time.

Finally, we highlight that we cannot identify the best option among the three tested *FLP*s in the first-stage for solving *2E-TD-LRP*. Indeed, on average the three versions of the heuristic showed similar performances on the tested instances.

Table 2 Results of the proposed approaches on the instances with $|N| = 20$

		B&C				FLP+TSP+LS				
ID	$\|C\|$	Ct	$T\%$	$D\%$	Rt	Ct	$T\%$	$D\%$	Rt	$Gap\%$
1	30	120	40.83	59.17	322	121	49.59	50.41	156	0.83
2	30	113	47.79	52.21	946	113	47.79	52.21	164	0.00
3	30	83	36.14	63.86	847	83	36.14	63.86	126	0.00
4	30	90	52.22	47.78	335	90	52.22	47.78	171	0.00
5	30	96	50.00	50.00	532	96	50.00	50.00	158	0.00
6	30	93	31.18	68.82	945	93	31.18	68.82	228	0.00
7	30	87	22.99	77.01	878	87	35.63	64.37	201	0.00
8	30	74	33.78	66.22	391	74	33.78	66.22	175	0.00
9	30	81	43.21	56.79	312	81	43.21	56.79	256	0.00
10	30	89	47.19	52.81	844	89	47.19	52.81	102	0.00
11	40	97	30.93	69.07	547	97	30.93	69.07	198	0.00
12	40	75	48.00	52.00	655	75	48.00	52.00	273	0.00
13	40	72	40.28	59.72	574	75	50.67	49.33	165	4.17
14	40	72	34.72	65.28	803	72	34.72	65.28	119	0.00
15	40	90	47.78	52.22	975	90	47.78	52.22	202	0.00
16	50	128	49.22	50.78	642	128	49.22	50.78	178	0.00
17	50	94	39.36	60.64	849	94	39.36	60.64	192	0.00
18	50	84	33.33	66.67	776	84	33.33	66.67	181	0.00
19	50	98	42.86	57.14	446	98	42.86	57.14	265	0.00
20	50	94	40.43	59.57	910	94	40.43	59.57	287	0.00
Mean		91.50	40.61	59.39	676.45	91.70	42.20	57.80	189.85	0.25

6 Conclusions

In this work we studied a two-echelon truck-and-drone location-routing problem *2E-TD-LRP* arising from the employment of an innovative delivery system composed by a truck, that acts as a mobile depot, and a fleet of drones. The problem has been modeled by a *MILP* formulation, solved through a B&C algorithm. Moreover, we also developed a two-stage solution method exploiting the underlying structure of the problem, consisting in an *FLP* and a *TSP*. Three variants of the proposed heuristic, differing in the kind of *FLP* solved at the first-stage, have been proposed. The computational results showed the applicability of the proposed methods. Moreover, the comparable percentages of truck and drone travelling times with respect to the overall delivery process time, show the benefits in terms of emission reduction that can be achieved when using low-emission and environmental friendly vehicles.

Future research perspectives may include extensions of the proposed approaches dealing with: more single-truck/multiple-drone systems; drones with different characteristics (i.e., endurance, speed); no-fly zones.

References

1. Agatz, N., Bouman, P., Schmidt, M.: Optimization approaches for the traveling salesman problem with drone. Transp. Sci. **52**(4), 965–981 (2018)
2. Boccia, M., Crainic, T.G., Sforza, A., Sterle, C.: Location-routing models for designing a two-echelon freight distribution system. Rapport technique, CIRRELT, Université de Montréal, 91 (2011)
3. Boccia, M., Crainic, T.G., Sforza, A., Sterle, C.: Multi-commodity location-routing: Flow intercepting formulation and branch-and-cut algorithm. Comput. Oper. Res. **89**, 94–112 (2018)
4. Boccia, M., Masone, A., Sforza, A., Sterle, C.: A column-and-row generation approach for the flying sidekick travelling salesman problem. Transp. Res. C Emerg. Technol. **124**, 102913 (2021)
5. Boccia, M., Masone, A., Sforza, A., Sterle, C.: An exact approach for a variant of the FS-TSP. Transp. Res. Procedia **52**, 51–58 (2021)
6. Bosona, T.: Urban freight last mile logistics—challenges and opportunities to improve sustainability: a literature review. Sustainability **12**, 8769 (2020)
7. Cary, N., Bose, N.: UPS, FedEx and Amazon gather flight data to prove drone safety, September 24, 2016. https://venturebeat.com/2016/09/24/ups-fedex-and-amazon-gather-flightdata-to-prove-drone-safety/
8. Chang, Y., Lee, H.: Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. Expert Syst. Appl. **104**, 307–317 (2018)
9. Cleophas, C., Ehmke, J.F.: When are deliveries profitable? Bus. Inf. Syst. Eng. **3**, 153–163 (2014)
10. Contardo, C., Iori, M., Kramer, R.: A scalable exact algorithm for the vertex p-center problem. Comput. Oper. Res. **103**, 211–220 (2019)
11. Cuda, R., Guastaroba, G., Speranza, M.G.: A survey on two-echelon routing problems. Comput. Oper. Res. **55**, 185–199 (2015)
12. Drexl, M., Schneider, M.: A survey of variants and extensions of the location-routing problem. Eur. J. Oper. Res. **241**, 283–308 (2015)
13. Ferrandez, S.M., Harbison, T., Weber, T., Sturges, R., Rich, R.: Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. J. Ind. Eng. Manag. **9**(2), 374–388 (2016)
14. Kille, T., Bates, P.R., Lee, S.Y.: Unmanned Aerial Vehicles in Civilian Logistics and Supply Chain Management. IGI Global, Hershey PA (2019)
15. Masone, A., Sforza, A., Sterle, C., Vasilyev, I.: A graph clustering based decomposition approach for large scale p-median problems. Int. J. Artif. Intell. **16**(1), 116–129 (2018)
16. Otto, A., Agatz, N., Campbell, J., Golden, B., Pesch, E.: Optimization approaches for civil applications of unmanned aerial vehicles (UAVS) or aerial drones: A survey. Networks **72**(4), 411–458 (2018)
17. ReVelle, C.S., Eiselt, H.A.: Location analysis: A synthesis and survey. Eur. J. Oper. Res. **165**(1), 1–19 (2005)
18. Salama, M., Srinivas, S.: Joint optimization of customer location clustering and drone-based routing for last-mile deliveries. Transp. Res. C Emerg. Technol. **114**, 620–642 (2020)
19. Sridharan, R.: The capacitated plant location problem. Eur. J. Oper. Res. **87**(2), 203–213 (1995)

A Heuristic Approach for the Human Migration Problem

Giorgia Cappello, Patrizia Daniele, and Federico Perea

Abstract In this paper, we present a network-based model for human migration in which a utility function is maximized. The resulting nonlinear optimization problem is characterized by a variational inequality formulation. Due to the high complexity of this problem, in order to efficiently solve realistic instances a heuristic method is proposed. The presented algorithms are tested and compared over a number of randomly generated instances.

Keywords Heuristics · Nonlinear programming · Variational inequality · Human migration network

1 Introduction

In 2019, international migrants worldwide comprise 3.5% of the global population, compared to 2.8% in 2000 (see [15]). At the end of 2019 almost 80 million people are forcibly displaced as a result of persecution, conflict, violence, human rights violations or events seriously disturbing public order (see [16]).

Their vulnerabilities may be exacerbated in crisis situations, as nowadays it is the case with the COVID-19 pandemic. This respiratory infectious disease emerged in China and quickly spread around the world posing enormous health, social, economic, environmental, and political challenges to the entire human population (see [1, 4, 7]).

G. Cappello · P. Daniele (✉)
Department of Mathematics and Computer Science, University of Catania, Catania, Italy
e-mail: giorgia.cappello@unict.it; daniele@dmi.unict.it

F. Perea
Department of Applied Mathematics II. Polytechnic School, University of Sevilla, Sevilla, Spain
e-mail: perea@us.es

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_14

Migration significantly contributes to economic and social development everywhere and is one of the main topics in the political debate worldwide (such as, migration will be key to achieve the Agenda 2030 Sustainable Development Goals (SDGs)).

The global pandemic has highlighted how important it is for the governments to work on the introduction and development of migration management policies arising from this global health crisis (see [13]).

In the last decades, some scientific literature has been devoted to this topic, and, for brevity, only few of them are given below.

In [10], Nagurney proves that, in the case of linear utility functions and fixed movement costs, the equilibrium conditions for the network equilibrium mode can be reformulated as the solution to an equivalent quadratic programming problem. In [5], Kalashnykov and Kalashnykova establish the equivalence of the equilibrium to a solution of an appropriate variational inequality problem. In [2], Cappello and Daniele present a multiclass migration model where the aim of each migration class is to maximize the attractiveness of the origin country and prove that the optimization model can be formulated in terms of a Nash equilibrium problem and a variational inequality. In [12], the authors demonstrate that, through policy interventions, in the form of subsidies, a system-optimum for a multiclass human migration network can be achieved, despite the migrants behaving in a user-optimized manner. The authors extend their work in [11], with the generalization of the inclusion of capacities associated with the migrant classes and locations.

1.1 Our Contribution and Organization of the Paper

In this paper, we recall the system-optimized models of human migration in [2]. In the objective function, as in [3], we take into account the changes in the utility functions of the multiple classes caused by the migratory flows and policies adopted by governments. Further, in determining the optimal flows, we consider the government policies a priori, thanks to a suitable coefficient influence vector w. Our aim is to find a system-optimized solution, which is a social optimum, in that an organization, such as the United Nations, maximizes the attractiveness of the origin countries with respect to the destination one, for each migration class and each pair of countries (or locations). Since the human migration model gives rise to a variational inequality, the optimal solution can be found by using exact methods, for example the projection method. However, sometimes, depending on the size of the problem, computational times of the exact methods are very long and, therefore, it is convenient to use a heuristic approach, which allows us to obtain good solutions, in much faster time. Note that, typically, the determination of good approximate solutions is enough in real applications (especially if referring to large problems). This is essentially due to a number of factors: many of the parameters present in real applications are estimates that can be affected by errors, for which it is not worth waiting too long to have a solution whose value (or feasibility) is uncertain. In some

cases, one is interested in having a possible solution for the problem in order to evaluate quickly work scenarios (operational contexts, integration of optimization algorithms in interactive decision support systems). In emergency situations, such as the current Covid-19 pandemic, for the Governments to make decisions rapidly it is crucial to have efficient solution methods. In the Government management of the migration phenomenon it may be interesting to evaluate, in real time, migration policies and the configurations of optimal migratory flows, while guaranteeing rights and welfare in every nation.

The rest of the paper is organized as follows. In Sect. 2, we recall the network based model and the associated variational inequality formulation. In Sect. 3, we outline a heuristic approach to solve realistic instances of the optimization problem proposed in this paper. The presented algorithm is tested and compared against an exact one over a number of randomly generated instances in Sect. 4.

2 The Mathematical Model

In this section we present a multiclass migration network model, based on the model by Cappello and Daniele (see [2]). We briefly describe the topology of the network depicted in Fig. 1, which is composed by n nodes, that are countries (or locations) where the different classes choose to start or end their migration, and m migration classes that move (or not) from one node of the network to another. Let's introduce the node set $N = \{1, \ldots, n\}$, the generic node is indexed by letters i and j and migration class set $M = \{1, \ldots, m\}$, the generic migration class is indexed by letter k. In each node i in the network there is an initial population for each migration class k, denoted by $\bar{p}_i^k$, which is assumed known and deterministic.

Let us introduce the following variables:

- $f_{ij}^k \geq 0$ is the flow of migration class k out of node i and into node j in the network. Let $f = (f_{ij}^k)_{i,j \in N;\, i \neq j;\, k \in M}$ the flow vector.
- $p_i^k \geq 0$ is the population of migration class k at node i, after the migration flow. Let $p = (p_i^k)_{i \in N;\, k \in M}$, the population vector.

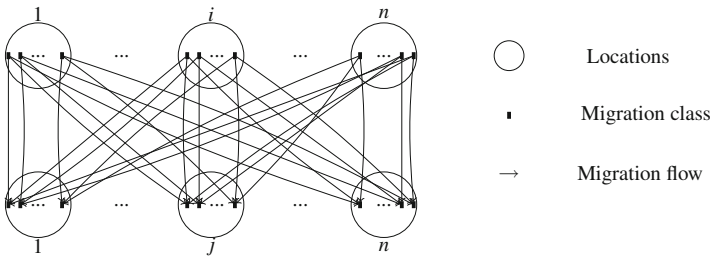


Fig. 1 Network structure of the multiclass human migration model

The volume of population of each class k at each node i, after the migration takes place, is given by the following flow conservation constraints:

$$p_i^k = \bar{p}_i^k - \sum_{\substack{j=1 \\ j \neq i}}^{n} f_{ij}^k + \sum_{\substack{j=1 \\ j \neq i}}^{n} f_{ji}^k \quad \forall\, i, k. \tag{1}$$

Besides, for each migration class k, the sum of the flows out of each node i must not exceed the initial fixed population in i, that is,

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} f_{ij}^k \leq \bar{p}_i^k, \ \forall\, i. \tag{2}$$

We denote the feasible set of (p, f) vectors as follows:

$$\mathbb{K} = \left\{ (p, f) \in \mathbb{R}_+^{mn+mn(n-1)} : \ (1),\ (2) \text{ hold } \forall\, i, j, k,\ i \neq j \right\}. \tag{3}$$

We now introduce the functions of the model (see [2] and [3] for more details on economic interpretation of the following functions):

- $u_i^k(p)$ and $v_j^k(p)$ denote the origin and destination utility functions, which capture the attractiveness from an economic and/or political and/or social point of view of the origin node i and of the destination node j, respectively, as perceived by a single individual of the migration class k.
- $w_{ij}^{k-}(p, f) \geq 0$ and $w_{ij}^{k+}(p, f) \geq 0$ are the influence coefficients for an individual of migration class k and for each possible migration route from node i to node j, which allow us to consider the migration policies implemented in the aforementioned nodes as a consequence of the utility changes when individuals of class k choose to migrate from node i towards node j, respectively.
- $f_{ij}^k w_{ij}^{k+}(p, f) \frac{\partial v_j^k(p)}{\partial p_j^k}$ and $-f_{ij}^k w_{ij}^{k-}(p, f) \frac{\partial u_i^k(p)}{\partial p_i^k}$ represent the possible variation of the utility functions due to a migratory movement in the network from location i to location j by class k and the consequent policies adopted by the governments in the nodes. Both variations are assumed to be concave.
- $c_{ij}^k(f)$ are the movement cost functions from i to j for the population class k and such costs are assumed to be convex and continuously differentiable.

For each migration class k, we define its *net utility function*, $U^k(p, f)$, as the sum of the attractiveness in i (which is given by the sum of the utility in i and its expected variation of utility function) to which both the attractiveness in j (which is given by the sum of the utility in j and its expected variation of utility function) and the transportation costs from location i to location j, are subtracted. Namely, the

following difference:

$$U^k(p, f) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \left(u_i^k(p) - f_{ij}^k w_{ij}^{k-}(p, f) \frac{\partial u_i^k(p)}{\partial p_i^k} - v_j^k(p) - f_{ij}^k w_{ij}^{k+}(p, f) \frac{\partial v_j^k(p)}{\partial p_j^k} - c_{ij}^k(f) \right) \tag{4}$$

for all $k \in M$, where $(p, f) \in \mathbb{K}$.

We also define the *total utility function*, $U(p, f)$, as follows

$$U(p, f) = \sum_{k=1}^{m} U^k(p, f), \tag{5}$$

where $(p, f) \in \mathbb{K}$.

In our multiclass human migration network system-optimization problem, the cognizant organization seeks to determine the optimal flows, as well as the optimal populations at each node in the network, and for all the migration classes. Hence, the optimization problem can be expressed as follows:

$$\text{Maximize } U(p, f), \tag{6}$$

where $(p, f) \in \mathbb{K}$.

Under the above assumptions, the objective function in (6) is concave and continuously differentiable and so, using the classical variational theory (see [6] and [9]), it is easy to prove that an optimal solution for (6), denoted by $(p^*, f^*) \in \mathbb{K}$, satisfies the following variational inequality (VI):

Find $(p^*, f^*) \in \mathbb{K}$ such that:

$$\frac{\partial U(p^*, f^*)}{\partial p} \times \left(p - p^*\right) + \frac{\partial U(p^*, f^*)}{\partial f} \times \left(f - f^*\right) \geq 0 \quad \forall (p, f) \in \mathbb{K}. \tag{7}$$

3 A Heuristic Approach

The complexity of the previous variational inequalities makes it too time consuming, or even impossible, to solve realistic instances. Therefore, in this section we propose a heuristic algorithm that allows us to find a good feasible solution in a reasonable amount of time.

Our optimization problem consists of (6).

The heuristic approach we propose consists of the following steps:

Step 1. *Initial population generation* ($\mathscr{P}$). For each component $\ell = 1, \ldots, mn(n-1)$ of the flow vector f, we generate n_t matrices with n_p rows and $mn(n-1)$ columns. In each matrix, each row represents one flow vector. These n_t matrices are generated in such a way that component ℓ is fixed for all rows (α in the pseudocode below). The other components of each row are found randomly in such a way that (3) is satisfied. Each row (each flow vector) is completed with the population vector p, obtained from equalities (1).[1] Each row of all the matrices is, now, a (p, f) vector. We evaluate each (p, f) vector and we keep the vector (p, f) that maximizes the objective function, among all $mn(n-1) \times n_t$ such (p, f) vectors generated. We denote that vector as (p_ℓ, f_ℓ). We repeat this process for all components $\ell = 1, \ldots, mn(n-1)$, and we store all (p_ℓ, f_ℓ) vectors, which will constitute the initial population set, denoted by $\mathscr{P}$.

Step 2. *Combination*. We randomly select three population vectors, (p^{r_1}, f^{r_1}), (p^{r_2}, f^{r_2}), $(p^{r_3}, f^{r_3}) \in \mathscr{P}$. Then, we generate a new vector, $(\bar{p}, \bar{f})$, which is a convex combination (p^{r_2}, f^{r_2}) and (p^{r_3}, f^{r_3}), as follows:

$$(\bar{p}, \bar{f}) = F(p^{r_2}, f^{r_2}) + (1 - F)(p^{r_3}, f^{r_3}),$$

where F is a real number randomly generated in $(0, 1)$. Note that, since $(\bar{p}, \bar{f})$ is a convex combination of $(p^{r_2}, f^{r_2}), (p^{r_3}, f^{r_3}) \in \mathbb{K}$, then $(\bar{p}, \bar{f}) \in \mathbb{K}$ too.

Step 3. *Selection*. We evaluate the vector $(\bar{p}, \bar{f})$: if it has an objective function value higher than (p^{r_1}, f^{r_1}), we include it in the population and remove the vector (p, f) with the minimum objective function value in the population. Update $\mathscr{P}$.

Step 4. Steps 2 and 3 are repeated for t_u seconds. Afterwards, the best solution in $\mathscr{P}$ is returned as the result.

Algorithm 1 shows a pseudocode of this heuristic.

4 Computational Experiments

In this section we assess the proposed algorithms over a benchmark of randomly generated instances. Each instance is solved by both the heuristic and the Projection-Contraction method (see [14]). Both algorithms were coded using Matlab and were run on a PC with 4 GB RAM, Asus Intel (R) Core (TM) i5-3317U CPU@1.10 GHz.

[1] From the flow conservation equations (1), we observe that the components of the population vector p can be expressed in terms of the flow vector f.

Algorithm 1 Pseudo code

procedure INITIAL POPULATION GENERATION
  $\mathscr{P} = \emptyset$
  for $\ell = 1 : mn(n-1)$ **do**
    for $s = 1 : n_t$ **do**
      Generate α randomly s.t. (2) is satisfied
      Generate randomly n_p flow vectors such that the ℓ-th component is α and (2) is satisfied.
      Complete each of the n_p flow vectors f with the corresponding population vector p, obtained from (1).
      Select the vector (p_s, f_s) with the highest objective function value.
    End For
    Select the vector (p_ℓ, f_ℓ) with the highest objective function value among the n_t (p_s, f_s) vectors.
    $\mathscr{P} = \mathscr{P} \cup \{(p_\ell, f_\ell)\}$.
  End For
procedure COMBINATION
  while time $< t_u$ **do**
    Select randomly three population vectors
    $(p^{r_1}, f^{r_1}), (p^{r_2}, f^{r_2}), (p^{r_3}, f^{r_3}) \in \mathscr{P}$
    F is randomly generated in (0, 1).
    Generate a vector $(\bar{p}, \bar{f}) = F * (p^{r_2}, f^{r_2}) + (1 - F) * (p^{r_3}, f^{r_3})$
    procedure SELECTION
      Evaluate the vector $(\bar{p}, \bar{f})$
      if $U(\bar{p}, \bar{f}) \geq U(p^{r_1}, f^{r_1})$ **then**
        $\mathscr{P} = \mathscr{P} \setminus \{(p^w, f^w)\}$ (the worst element)
        $\mathscr{P} = \mathscr{P} \cup \{(\bar{p}, \bar{f})\}$
      end
  end
Return the vector with the highest objective function value

4.1 Instance Generation

We now explain how the benchmark of random instances has been generated. We first define four different network configurations for our migration problem. Afterwards, each of these four configurations will be replicated 50 times with randomly generated data. So, in total we have $4 \times 50 = 200$ different instances. The first 10 instances of each configuration will be used for calibrating the different parameters of our heuristic algorithm, namely as test set, while the other 40 instances of each configuration will be used as evaluation set. In both the sets we compare the performance of our heuristic with respect to the projection method.

4.1.1 Configurations

In this section we explain the four illustrative configurations used for our experiments. All of them are obtained from the topology of the migration network analysed in the illustrative numerical example shown in [2]. We consider a single migration class ($M = \{1\}$), as in [2] and in [5] in which a realistic numerical

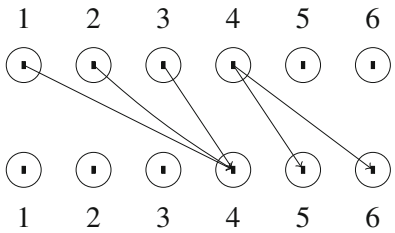


Fig. 2 Configuration 1

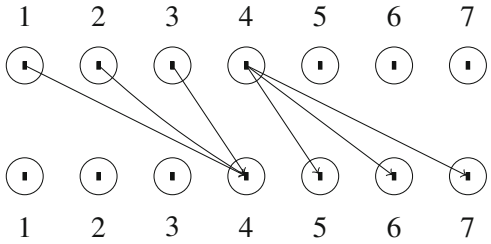


Fig. 3 Configuration 2

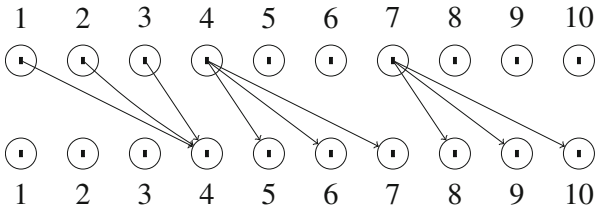


Fig. 4 Configuration 3

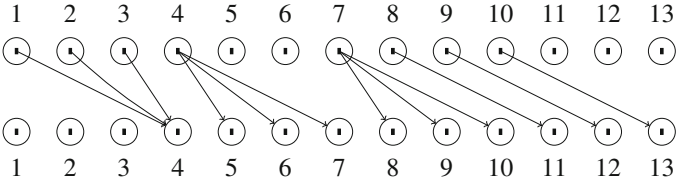


Fig. 5 Configuration 4

example with a unique class moving within Mexico is analysed. In order to have larger instances, we progressively increase the number of nodes in the migratory network. We have chosen that the total number of nodes varies from 6 to 13, as depicted in Figs. 2, 3, 4, and 5, respectively.

4.1.2 Random Generation

Each configuration is replicated 50 times, by randomly generating their input data as explained below. Since in all the configurations M only consists of one single migration class, we omit the superscript notation.

For each configuration in Sect. 4.1.1, the origin and destination utility functions and the transportation cost functions were chosen polynomial as follows:

$$u_i(p_i) = -\text{coef}_i * (p_i)^2, \quad v_j(p_j) = \text{coef}_j * (p_j)^2; \quad \forall\, i,\ j \in N,$$
$$c_{ij}(f_{ij}) = \text{coef}_{ij} * f_{ij}; \quad \forall\, i, j \in N;\ i \neq j,$$

where the coefficients were integer generated randomly as follows:

$$\text{coef}_i \in \{1, \ldots, 15\}, \quad \text{coef}_j \in \{1, \ldots, 15\}, \quad \forall i, j \in N;$$
$$\text{coef}_{ij} \in \{400, \ldots, 1000\} \ \forall\, i, j \in N,\ i \neq j.$$

The initial populations of each node are randomly generated using an integer uniform distribution as follows:

$$\bar{p}_i \in \{3000000, \ldots, 11000000\}, \ \forall i \in N.$$

Regarding $w_{ij}^{\pm}(p, f)$, we assume they are constant for all (p, f) but different for each $i, j \in N, i \neq j$, with values randomly generated in [0, 1].

4.1.3 Calibration

In order to choose the optimal values for the parameters n_p and n_t of the heuristic algorithm, in this section we perform an analysis of experiments to find such optimal values. Each parameter is tested over the following values:

- $n_p \in \{100, 150, 200, \ldots, 1000\}$, having in total 19 different values.
- $n_t \in \{150, 200, 250, 300, 350\}$, having in total 5 different values.

The lowest values tested for the parameters were chosen considering that, based on initial experience, good solution quality was not obtained for any lower values. The largest values tested for the parameters were chosen so that the CPU times would not be too large.

As explained before, the first 10 instances of each configuration will constitute the calibration set. Each of these instances has been run for every possible combination of parameters. Therefore, in this calibration phase we accumulate a total of $19 \times 5 \times 10 = 3800$ experiments.

We analyse the results with a powerful statistical tool: the Analysis of Variance (ANOVA) technique. The interested reader is referred to [8] for a complete review on this tool. Both algorithm parameters (n_p, n_t) are considered as factors in a

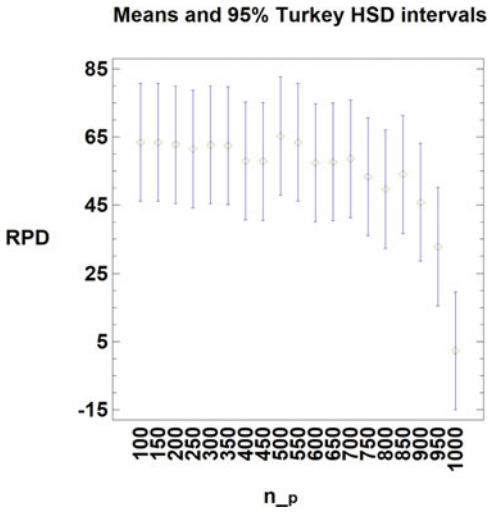


Fig. 6 Mean plot for n_p in the ANOVA

full factorial experiment. The considered response variable is the Relative Percent Deviation (RPD), defined for each instance and each combination of parameters as follows:

$$RPD = 100\frac{Z_{exact} - Z_{comb}}{Z_{exact}}, \tag{8}$$

where Z_{exact} is the value of the solution obtained with the exact method, and Z_{comb} is the value of the solution found by the heuristic with a particular combination of parameter values tested.

The complete ANOVA results are not given for simplicity. However, we do show the means plot and Tukey's Honest Significant Difference (HSD) 95% confidence intervals in Fig. 6 for parameter n_p, and in Fig. 7 for parameter n_t.

We observe that, for n_p, the best results (lower RPD) are obtained for the largest value of this parameter: $n_p = 1000$. We also observe that the differences between this value and the other values are always statistically significant, as the corresponding HSD intervals do not overlap, except for $n_p = 950$.

Regarding n_t, we do not observe statistically significant differences between the different values tested, as all HSD intervals do overlap. However, the best combination of parameters in terms of average values is $n_t = 350$.

In the ANOVA analysis we also observed that there is no significant interaction between the two parameters tested.

Therefore, the chosen combination of parameters for our heuristic algorithm is $n_p = 1000, n_t = 350$. In the following section we compare our heuristic algorithm with the exact method, using the aforementioned values for the two parameters in the heuristic. All the results, about execution time and RPD value, are shown in the next section.

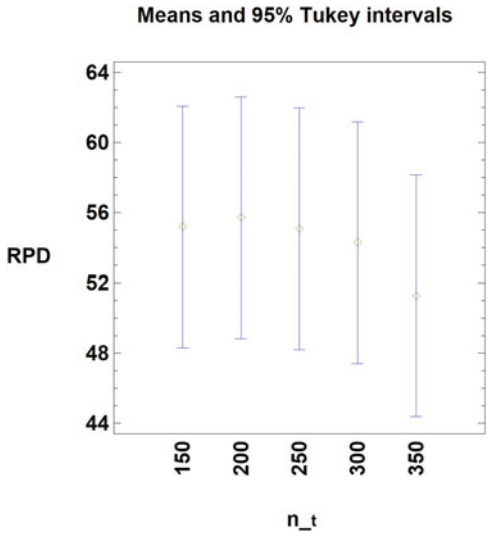


Fig. 7 Mean plot for n_t in the ANOVA

4.2 Evaluation

In this section we show the comparison between the heuristic algorithm, with the parameters obtained in the calibration section, and the exact algorithm. For each instance, we store the following data:

- *i_b*: the iteration in which the solution with the highest value of the objective function is found;
- *t_b*: the time in which the best solution is found;
- *tot_i*: the total number of iterations that the algorithm performs for Steps 2–4;
- *tot_t*: the total execution time of the heuristic including Steps 1–4;

For all instances, steps 2–4 of the algorithm are repeated for 30 s.

We report here, for brevity, the average of the aforementioned data:

- Config_1: i_b = 3366.78, t_b = 0.17, tot_i = 608891.3, tot_t = 72.93
- Config_2: i_b = 216.06, t_b = 0.06, tot_i = 112865.9, tot_t = 90.60
- Config_3: i_b = 8875.46, t_b = 0.52, tot_i = 380406.9, tot_t = 150.03
- Config_4: i_b = 1062.16, t_b = 0.68, tot_i = 39022.76, tot_t = 244.98

The average is calculated between the values obtained from the first 10 instances with the best combination of parameters (n_p, n_t) (defined in the previous section), and the instances of the evaluation set.

We observe how, even if the heuristic runs for over a minute (72 s for the first configuration, 90 for the second, 150 for the third, and 244 for the fourth), the best solution is always found in less than a second, on average (0.17 seconds for the first configuration, 0.06 for the second, 0.52 for the third configuration, and 0.68 for the

fourth). This means that our heuristic finds good solutions (as will be proven next) in very short computational times. We now compare the performance of the two methods (heuristic algorithm and exact algorithms). In Table 1 we show a summary of such a comparison.

For each instance and each configuration, we store the following data:

- $Time_H$: the execution time of the heuristic algorithm;
- $Time_E$: the execution time of the exact algorithm;
- RPD: the relative percent deviation between the values of the solutions obtained by the heuristic method and by the exact method, computed using the formula (8)

We observe how the heuristic method finds solutions very close to the optimal in quicker CPU times than the exact method. Besides, the heuristic method is expected to be able to yield good feasible solutions in larger instances where the exact method is not able to give a solution. Further, from Table 1 it is possible to remark that, moving from Configuration 1 to Configuration 4, the RPD mean value gradually increases, which is strictly connected to the time limit of 30 s in steps 2–4 of the algorithm (1), for all instances, regardless the number of nodes of the configurations.

5 Conclusion

In this paper we have examined a network based human migration problem, obtaining a nonlinear optimization model, since all the classes of population aim at maximizing their own utility.

In order to solve real cases, we have applied a heuristic algorithm and compared the solutions with the optimal ones obtained using the projection-contraction method.

Extensive computational experiments have shown the efficiency of the proposed algorithm since it allows us to find solutions very close to the optimal in quicker CPU times than the exact method. Further, such an algorithm is expected to be able to yield good feasible solutions in larger instances where the exact method is not able to give a solution.

Future studies could be developed applying the same technique and with suitable changes to other optimization models inspired by a pandemic, a natural disaster or to new multiclass network models of human migration were policy interventions are included, in line with the latest advances in literature.

Table 1 Numerical results

	Config. 1			Config. 2			Config. 3			Config. 4		
	Time_E	Time_H	RPD(%)	Time_H	Time_E	RPD(%)	Time_E	Time_H	RPD(%)	Time_E	Time_H	RPD(%)
Inst.1	4.14	50.11	0.04	1677.43	56.99	1.71	1769.03	154.99	1.6	150.68	243.61	2.75
Inst.2	13574.65	74.9	0.4	1.1	57.02	0.18	1714.44	156.63	7.37	2194.83	250.52	6.9
Inst.3	336.6	50.09	0.33	1203.53	57	1.19	1471.93	146.76	1.99	1625.04	244.24	2.46
Inst.4	4.4	50	0.24	2314.72	57.35	2.55	230.45	146.97	9.67	944	233.22	4.14
Inst.5	285.39	50.3	0.03	153.34	57.05	0.25	399.01	138.28	0.21	320.64	244.55	0.82
Inst.6	506.05	49.83	0.75	252.86	57.09	0.03	1.25	140.59	3.56	3.84	244.45	0.12
Inst.7	180.97	50.37	0.24	1145.78	57.13	2.14	171.45	140.44	3.02	384.39	248.44	1.47
Inst.8	395.72	50.28	0.11	1.45	57.09	1.36	254.82	139.18	4.56	308.98	121.9	3.34
Inst.9	335.26	50.18	0.73	1.53	57.14	0.59	421.31	140.76	0.61	706.35	121.56	6.13
Inst.10	338.19	50.32	0.44	2.1	57.08	0.15	2229.46	139.98	12.46	703.81	121.47	2.95
Inst.11	583.44	77.87	0.34	4.74	97.01	0.65	516.14	142.16	1.47	358.97	257.67	1.05
Inst.12	263	77.33	0.71	919.69	97.06	0.35	1504.82	142.21	3.83	582.88	258.62	1.63
Inst.13	350.34	78.31	1.07	1722.51	103.3	1.85	227.8	143.93	3.17	403	238.03	1.54
Inst.14	421.94	78.65	0.48	518.42	96.76	1.64	418.06	143.7	0.1	443.16	240.05	4.64
Inst.15	391.82	77.9	0.39	3.95	97.3	0.14	6.18	142.77	1.07	466.17	238.53	0.6
Inst.16	1032.61	77.14	0.63	1.09	97.89	1.31	481.27	149.77	2.03	2440.07	246.28	6.12
Inst.17	7.56	77.38	0.45	1168.8	109.34	1.13	263.21	156.36	1.03	2188.77	245.75	4.94
Inst.18	1.73	77.02	0.1	1.19	109.45	0.34	398.13	145.76	3.18	2753.08	255.3	4.84
Inst.19	908.85	77.23	0.37	314.37	104.91	0.8	1965.89	142.26	0.34	12783.87	247.91	3.4
Inst.20	326.95	77.88	0.59	1211.18	105.27	0.75	259.73	143.4	4.24	589.94	254.77	6.32
Inst.21	5405.62	79.48	6.71	156.46	99.05	0.17	491.69	142.5	1.59	664.48	249.16	3.81
Inst.22	1375.35	77.64	0.72	491.76	97.36	2.42	982.78	142.36	3.38	1341.97	246.6	2.43
Inst.23	307.56	77.79	0.41	377.35	102.4	0.61	475.61	142.35	2.25	366.61	248.75	0.38

(continued)

Table 1 (continued)

	Config. 1			Config. 2			Config. 3			Config. 4		
	Time_E	Time_H	RPD(%)	Time_H	Time_E	RPD(%)	Time_E	Time_H	RPD(%)	Time_E	Time_H	RPD(%)
Inst.24	321.53	78	0.5	491.42	97.91	0.36	1.47	141.02	1.16	739.96	260.26	6.76
Inst.25	369.85	76.99	0.66	966.62	101.42	2.43	353.99	141.49	5.17	2885.67	255.93	8.64
Inst.26	269.16	76.93	0.64	344.08	99.31	0.55	2585.98	141.53	0.1	1438.41	262.54	3.52
Inst.27	329.48	77.15	0.44	232.45	98.33	0.3	7.06	141.4	0.86	1898.75	250.09	1.84
Inst.28	451.02	77.68	0.23	375.69	96.91	0.68	1.26	141.44	1.28	1411.3	252.39	0.97
Inst.29	483.69	77.47	2.03	291.54	97.59	0.09	353.22	142.78	0.91	734.36	250.99	8.13
Inst.30	697.49	77.36	0.18	351.98	104.44	1.76	386.05	145.74	1.95	468.42	252.73	4.07
Inst.31	551.64	77.2	0.95	341.12	98.95	0.6	3303.19	141.71	3.96	239.5	254.65	2.46
Inst.32	151.48	78.74	0.6	0.99	97.24	0.5	593.06	142.79	3.9	381.89	257.58	3.25
Inst.33	39.41	78.96	2.11	1343.27	98.71	2.13	617.33	141.59	0.62	2066.21	249.21	3.62
Inst.34	0.81	78.67	1.12	1127.69	97.24	0.48	2895.08	141.58	0.1	1407.19	272.15	7.04
Inst.35	0.52	77.78	0.42	1.73	96.62	0.27	383.66	141.47	8.18	669.51	313.31	5.03
Inst.36	1368.27	79.45	0.25	399.43	96.59	3.3	347.44	141.35	1.63	2967.46	259.4	11.78
Inst.37	0.73	78.08	0.13	1162.35	96.42	0.54	388.3	141.62	2.85	261.32	266.21	2.84
Inst.38	2058.2	78.22	0.93	1.7	97.7	0.21	1892.04	141.77	0.57	3967.92	249.28	1.27
Inst.39	728.31	77.59	0.01	258.43	96.71	0.09	399.81	141.48	1.46	633.66	260.08	2.18
Inst.40	503.04	77.44	0.54	1023.85	98.13	0.25	312.74	141.78	0.78	312.58	251.97	4.79
Inst.41	287.07	78.18	0	1923.65	97.05	2.33	906.41	141.39	4.18	446.01	250.56	2.47
Inst.42	330.42	79.13	0.15	231.64	97.39	0.8	3.5	147.2	1.07	2.37	255.2	0.95
Inst.43	492.67	77.87	3.56	413.89	97.48	0.74	1.09	143.52	0.59	525.15	256.65	6.74
Inst.44	0.89	77.93	1.35	268.12	96.81	2.6	1.48	143.56	0.69	344.2	254.11	1.26
Inst.45	513.6	78.34	1.83	1082.08	96.81	1.48	683.31	148.44	3.7	733.18	250.75	3.73

Inst.46	245.4	77.93	0.19	295.18	97.01	1.78	426.64	141.91	6	1246.74	252.69	5.69
Inst.47	505.59	79.59	0.86	339.39	98.8	0.29	437.3	142.09	0.5	1815.37	252.26	12.78
Inst.48	1149.37	78.12	1.24	4.9	96.37	0.42	1174.04	253.25	1.56	1174.04	253.25	1.56
Inst.49	555.14	79.92	0.39	1088.2	96.27	0.86	343.44	251.86	1.53	343.44	251.86	1.53
Inst.50	177.67	78.23	0.1	0.33	98.15	1.81	822.6	251.68	4.98	822.6	251.68	4.98
Average	798.41	72.94	0.75	560.14	90.61	1	725.52	150.03	2.66	1253.25	244.98	3.85

Acknowledgments The research was partially supported by the research project "Programma Ricerca di Ateneo UNICT 2020–22 Linea 2-OMNIA" of Catania. This support is gratefully acknowledged.

References

1. Ali, I., Alharbi, O.M.: COVID-19: disease, management, treatment, and social impact. Sci. Total Environ. **728**, 138861 (2020)
2. Cappello, G., Daniele, P.: A variational formulation for a human migration problem. In: Advances in Optimization and Decision Science for Society, Services and Enterprises, pp. 185–195. Springer, Cham (2019)
3. Cappello, G., Daniele, P., Nagurney, A.: A system-optimization model for multiclass human migration with migration costs and regulations inspired by the Covid-19 pandemic. Minimax Theory Appl. **6**(2), 281–294 (2021)
4. Chakraborty, I., Prasenjit M.: COVID-19 outbreak: migration, effects on society, global environment and prevention. Sci. Total Environ. **728**, 138882 (2020)
5. Kalashnikov, V., Kalashnykova, N., Rojas, R.L., MuÑos, M.M., Uranga, C., Rojas, A.L.: Numerical experimentation with a human migration model. Eur. J. Oper. Res. **189**(1), 208–229 (2008)
6. Kinderlehrer, D., Stampacchia, G.: An Introduction to Variational Inequalities and Their Applications, vol. 31. SIAM (1980)
7. Martin, A., Markhvida, M., Hallegatte, S., Walsh, B.: Socio-economic impacts of COVID-19 on household consumption and poverty. Econ. Disasters Clim. Change **4**(3), 453–479 (2020)
8. Montgomery, D.C., Runger, G.C.: Applied Statistics and Probability for Engineers. Wiley, London (2010)
9. Nagurney, A.: Network Economics: A Variational Inequality Approach, vol. 10. Springer, Berlin (1998)
10. Nagurney, A.: A network model of migration equilibrium with movement costs. Math. Comput. Model. **13**(5), 79–88 (1990)
11. Nagurney, A., Daniele, P., Cappello, G.: Capacitated human migration networks and subsidization. In: Dynamics of Disasters—Impact, Risk, Resilience, and Solutions, vol. 169, pp. 195–217. Springer, Cham (2020)
12. Nagurney, A., Daniele, P., Cappello, G.: Human migration networks and policy interventions: bringing population distributions in line with system optimization. Int. Trans. Oper. Res. **28**(1), 5–26 (2021)
13. OECD: Managing international migration under COVID-19 (2020). http://www.oecd.org/coronavirus/en/
14. Solodov, M.V., Tseng, P.: Modified projection-type methods for monotone variational inequalities. SIAM J. Control Optim. **34**(5), 1814–1830 (1996)
15. United Nation (UN), Department of Economic and Social Affairs (DESA), Population Division: International Migrant Stock 2019, Documentation (2019)
16. UN Refugee Agency(UNHCR): Global Trends Forced Displacement in 2019, Report (2019)

In-store Picking Strategies for Online Orders in Grocery Retail Logistics

Xiaochen Chou, Dominic Loske, Matthias Klumpp, Luca Maria Gambardella, and Roberto Montemanni

Abstract Customers shifting from stationary to online grocery shopping and the decreasing mobility of an ageing population pose major challenges for the stationary grocery retailing sector. To fulfill the increasing demand for online grocery shopping, traditional bricks-and-mortar retailers use existing store networks to offer customers click-and-collect services. The current COVID-19 pandemic is substantially accelerating the transition to such a mixed offline/online model, and companies like the one behind this study are facing the urgent need of a re-design of their business model to cope with the change. Currently, a majority of the operations to service online demand consists of in-store picker-to-parts order picking systems, where employees go around the shelves of the shop to pick up the articles of online orders. The optimization of such operations is entirely left to the experience of the staff at the moment. Since in-store operations are a major cost-driver in retail supply chains, this paper proposes optimization ideas and solutions for these in-store operations. With experimental simulations run on a real store with real online orders, we show that a simple optimization tool can improve the situation substantially. The method is easy to apply and adaptable to stores with complex topologies.

Keywords In-Store Picking · Grocery Retailing · Optimization · Traveling Salesman Problem

X. Chou · L. M. Gambardella
Università della Svizzera Italiana, Lugano, Switzerland
e-mail: xiaochen.chou@usi.ch; luca.gambardella@usi.ch

D. Loske
Universidad Católica San Antonio de Murcia, Guadalupe, Spain
e-mail: dominic.loske@fom-net.de

M. Klumpp
Georg-August-University of Göttingen, Göttingen, Germany
e-mail: matthias.klumpp@uni-goettingen.de

R. Montemanni (✉)
University of Modena and Reggio Emilia, Reggio Emilia, Italy
e-mail: roberto.montemanni@unimore.it

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_15

1 Introduction

The ongoing digital transformation, demographic change, and modifying customer requirements are the major challenges for the European food retailing sector at the beginning of the 2020s [3]. First, the digital transformation primarily affects the relationship between the end-customer and the food retailer itself. Digitalization is causing sales channels to differentiate and converge at the same time as customers use different shopping options in parallel and expect a seamless shopping experience across channels [13]. The transformation goes from the choice of sales channels to different payment systems [2]. Second, on the one hand demographic change is leading to a decreasing European population. On the other hand, and more relevant for grocery retailers, the ageing population is less mobile, which raises the necessity of expanding stationary retail store networks or investing in home delivery solutions. Finally, the demographic change decreases the number of persons per household and, consequently, product sizes and average number of items per purchase. Third, increasing home consumption and governmental measures on social distancing caused by the global COVID-19 pandemic pose one of the greatest changes in customer behaviour that the retailing sector experienced during the last decades [19, 22].

To meet these challenges, a majority of the grocery retailers are operating several forms of online grocery businesses [11]. Exploring the design of omnichannel operations, A classification proposed in paper [24] identifies three typologies [24]: First, an integrated distribution center for online and offline orders that must enable both bulk and single unit picking and delivery. Second, distribution centers are exclusively utilized to fulfill online orders. Third, stationary grocery retailers use their traditional brick-and-mortar structures for online order fulfillment, especially on the store-level. In this work we consider the latter model, which requires less monetary invested to be implemented and allows a quicker transition to a mixed offline/online business model. In this context, stores can pick orders click-and-collect operations, according to buy-online-pick-up-in-store (BOPS) concepts. Online orders are picked by the store staff.

While buy-online-pick-up-in-store concepts are already well examined from a marketing perspective, e.g., omnichannel promotions [18], customers' product choices [15], pricing strategies [16], and the impact of service quality [17], the optimization of in-store logistics operations is hardly examined. From a retail logistics perspective, manual picker-to-parts order picking systems in warehouses have been identified as laborious and cost-intensive tasks [6] where human learning [12], human perception [8], order batching [20], and storage assignment [1, 7] has received considerable attention in the last years. While process and picking automation in retail warehouses seem to be a promising concept for the future [4, 5], in-store operations in grocery stores will continue to rely on the human workforce.

Although these picking operations are comparable to the picker-to-parts order picking systems in traditional warehouses, differences regarding in-store layouts (restricted sizes and areas) as well as huge variations of product placements on

the store-level make in-store picking to a labor- and knowledge-intensive field that has received little attention in recent years. On the other hand, in-store operations are a major cost driver for retail supply chains [10]. And as the average number of item per purchase decreases, in-store operations will become more detailed and complex. Therefore, our research question is: How can in-store picking operations in for buy-online-pick-up-in-store concepts of omnichannel retailing be optimized? The remainder of this paper is structured as follows: In Sect. 2, we introduce an in-store operational optimization based on the BOPS model considered. Experimental simulation results in a real store are presented and discussed in Sect. 3. The conclusions are drawn in Sect. 4.

2 In-store Operational Optimization

In the BOPS model considered, grocery retailers take the online orders made by customers, and serve these orders as normal in-store orders. An employee of the store goes through the shelves to pick up the articles selected by the online customer in his/her order. A batch of orders from different customers can also be picked by one employee at the same time. This business model is typical when a traditional retailer has the desire to open to the online market, without however investing extra money and manpower to change the modus-operandi of a well-established bricks-and-mortar business. It is sustainable as long as the percentage of online orders over the total ones does not exceed certain threshold. In such a context, an in-store operational optimization approach of great interest emerges.

Given an online order, an employee has to collect all the articles through the store. Most retailers, like the one under investigation, have currently developed no optimization relative to this process. The employee in charge usually receives a list in which the articles are presented in alphabetic order. In such a case, an experienced employee normally takes advantage of the knowledge of the store topology and self-organize a time- and distance-saving path through the store after a preliminary exam of the list. In actual operations, the online orders pick-up tasks are often assigned to part-time workers with marginal experiences. These less expert figures tend to be unable to efficiently organize the pick-up task until they accumulate some experience. Both scenarios however would strongly benefit from a simple optimization tool that is able to sequence the orders in a way to minimize the travel distance to fulfill the order within the time required to conclude the task.

The optimization tool works in the following way. Knowing the layout of the store, the positions of the articles can be categorized by different zones. Each zone has multiple articles in the same category, and by visiting each zone, the employee collects all the articles on the list. Articles in a same category are normally in a same zone, although it can happen that a same category spreads across contiguous zones. With this information, optimizing the pick-up missions turns to visiting different zones in the store only once. In such a scenario, finding the optimized sequence of pick for an order boils down to a classic (open) traveling salesman problem (TSP,

see [9, 23]), where the items to pick up (with the position of their own zones) map to the cities, the entry point is the initial depot and the cash desks' location is the final depot. The problem is therefore modeled by a graph $G = (V, A)$ where V is the set of nodes to visit, including the entry point, the articles to pick (associated with a zone index), and the cash desks. A is the set of arcs which connect every couple of nodes in V.

We define t_{ij} as the travel time between nodes $(i, j) \in A$, with the artificial setting $t_{\alpha\beta} = 0$, being α an artificial node representing the entrance of the store and β an artificial node representing the cash desks' location. Since each node represents an article to pick (at a given zone), the travel time between articles in the same zone is set to zero. The pick-up time for an article at zone j is directly added to the travel times t_{ij} for each zone i. The entrance and the cash desks are always the first and last nodes when we calculate with the list. In this way, the problem becomes a classic TSP problem.

For the model we define a variable $x_{ij} = 1$ if the optimal path goes through the arc from node i to node j, 0 otherwise. With these elements, we can adopt a classic TSP formulation for our problem as follows:

$$\min \quad \sum_{i \in V} \sum_{j \in V} t_{ij} x_{ij} \tag{1}$$

$$\sum_{i \in V, i \neq j} x_{ij} = 1 \qquad j \in V \tag{2}$$

$$\sum_{j \in V, j \neq i} x_{ij} = 1 \qquad i \in V \tag{3}$$

$$\sum_{i \in Q} \sum_{j \in Q, j \neq i} x_{ij} \geq 1 \qquad \forall Q \subseteq V, |Q| \geq 2 \tag{4}$$

$$x_{ij} \in \{0, 1\} \qquad i, j \in V, i \neq j \tag{5}$$

where (1) implies the minimization of the overall travel time, (2) and (3) impose that each node is considered is touched by the path, and (4) are the cutset form of the subtour elimination constraints, and forbid solutions with subcycles. Constraints (5) define the domain of the variables.

Similar application of the TSP to the retailer domain can be found in [21], where a mobile app aiming at suggesting customers the shortest path within collaborating shops from a given shopping list is described. Another study is presented in [14], relating the path followed by customers (against the potential shortest path) and the characteristics of the customers' shopping behavior. Both applications are customer-oriented design, while in this work we seek for in-store operational optimization for the shops. The main difference is that an in-store customer may self-organize the path in a partial optimized way knowing what he/she wants to buy (for example, collecting all items in need when arriving at a certain zone) and has relatively low requirement for efficiency. While an in-store employee under pick-up mission needs

to go through the store in an effective and efficient way to collect all items from the given shopping list. In such a situation, there will be great gains if the employee receives an optimized list.

3 Experimental Simulation

The study is carried out in a real store in Germany. The layout of the store is presented in Fig. 1, with numbers corresponding to areas/zones containing products of different categories. Zone 1 includes the entrance and Zone 14 the cash desks, that are the fixed starting point and destination in our model. The travel times match the real-world travel times. A pick-up time per article of 5 s is considered for all the zones (it is however technically possible to make it zone-dependent) and is added to the travel time as explained in Sect. 2. In this section, we consider 25 real online orders received by the store. The number of articles of each shopping lists varies from 20 to 50, with average of approximately 34 and standard deviation of approximately 7. In the simulation, we will compare the total time required to pick up all the items when pickers receive shopping lists ordered in different ways.

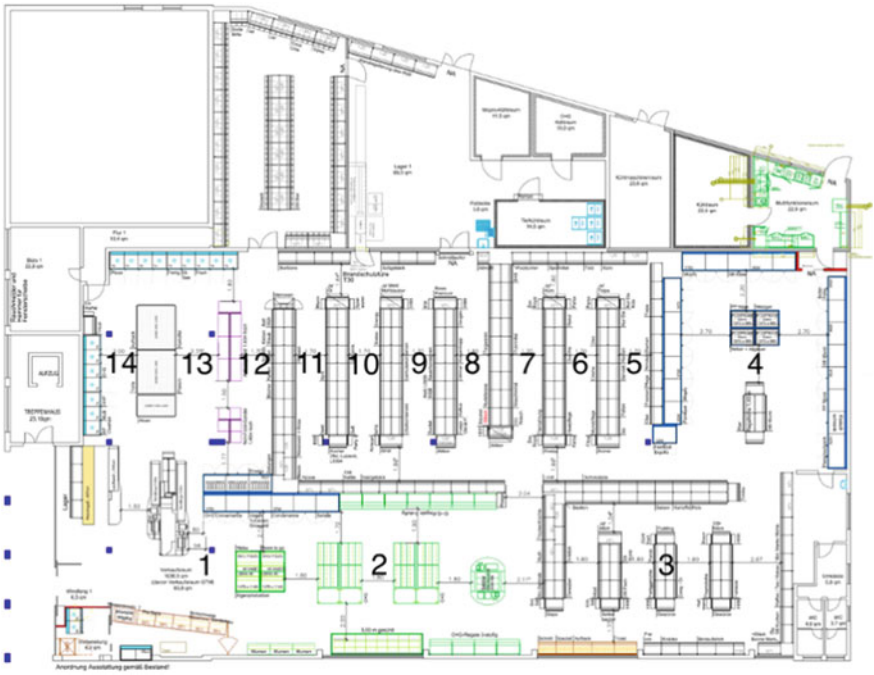


Fig. 1 Layout of a real store with numbers corresponding to areas containing products of different categories

The original shopping lists are in alphabetic order. As explained in Sect. 2, for an inexperienced employee, there will be a lot of back and forth moves with such a list. In the simulation, we suppose that an inexperienced employee collects the articles one by one following the order on the list, while an experienced employee collects all the articles on the list relative to a certain zone once arrived at that zone. In a third case, the employee (irrelevant of experienced or inexperienced) will be given a fully optimized list following the optimal TSP solution (being out users the employees, a visual map of with the path is judged superfluous and not provided, although this could be possible). Since the travel times between different articles in a same zone are considered as zero, and only pick-up times are taken into account, in this third case the employee is suggested to collect all the articles on the list relative to a certain zone once arrived at a that zone. To summarize, in the simulation the employee receives the shopping list(s) ordered in three different ways:

1. in alphabetic order
2. grouped by category according to the alphabetic order, but without optimizing the order of the categories
3. ordered according to the optimal solution of the TSP

Experimental simulations suggest that the gain of case 3 over 2 normally varies from 0% to 29%, while the gain of case 3 over 1 can be as high as over 70% in some extreme scenarios. Here we show two extreme cases as examples.

The first example we analyze in details is a shopping list containing articles all from two zones very close to each other. After remapping the list in alphabetic order to the numbers of zones, the shopping list contains 20 articles with nothing to be collected at the entrance and cash desks: {(1), 10, 11, 11, 10, 11, 11, 11, 11, 10, 11, 11, 11, 10, 11, 11, 11, 11, 10, 10, 11, (14)}. It takes 281 s to finish the pick-up when case 1 in considered. In case 2, the list is reordered as {(1), 10, . . . , 10, 11, . . . , 11, (14)}, that saves 74 s of travel time moving back and forth between zone 10 and 11. Case 3 overlaps with case 2 in such a list, thus no improvement is obtained.

The second example has more variants of articles located in different zones. The alphabetic shopping list contains 23 articles with nothing to be collected at the entrance and cash desk: {(1), 2, 10, 4, 6, 10, 6, 5, 2, 3, 2, 3, 3, 2, 7, 2, 2, 6, 2, 6, 3, 9, 8, 2, (14)}. It takes 828 s to finish the whole pick-up mission according to case 1. In case 2 the shopping list is transformed to {(1), 2, . . . , 2, 10, 10, 4, 6, . . . , 6, 5, 3, . . . , 3, 7, 9, 8, (14)} when handled by an experienced employee, who takes 350s to complete the mission. Her/his travelling path is shown in Fig. 2. We can observe that there are still some back and forth walking. In case 3, the employee receives the shopping list ordered according to the optimal TSP solution:{(1), 2, . . . , 2, 3, . . . , 3, 4, 5, 6, . . . , 6, 7, 8, 9, 10, 10, (14)}. In this case it takes only 247s to complete the pick-up task. Figure 3 shows the travel path of the employee, that is characterized by a consistent direction for each move, leading to higher efficiency.

From the simulations carried out in this real store, we can observe that both inexperienced (case 1) and experienced employee (case 2) can benefit from a shopping list filled in according to the optimal TSP solution. It is worth noting that the layout of this store is actually rather simple and direct. The in-store operations

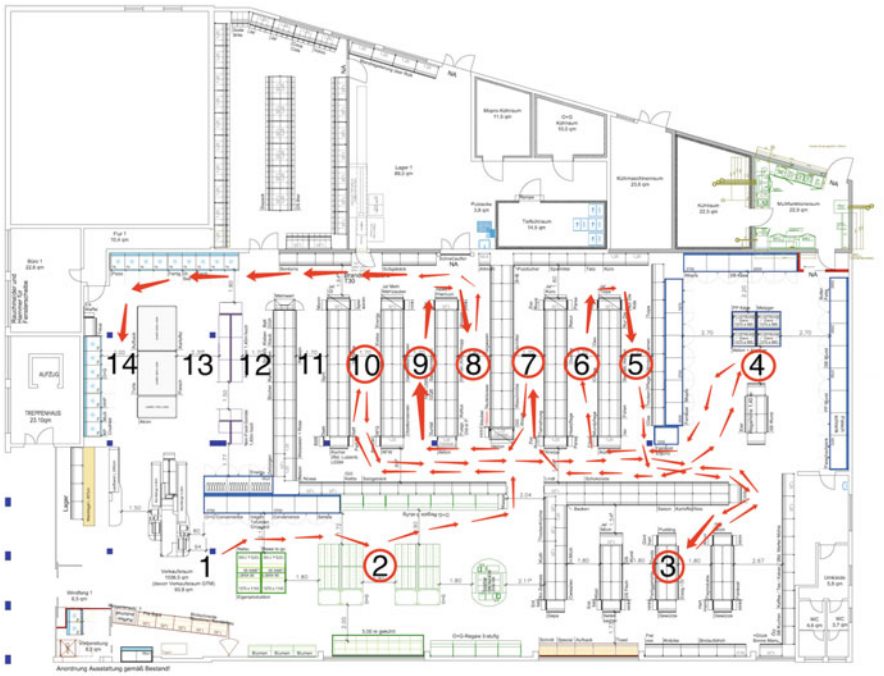


Fig. 2 Travelling path of a pick-up task completed by an experienced employee

optimization we proposed would provide an even strong benefit in stores with more complex layouts. Besides, it is also possible to pick up multiple orders in parallel by one employee with a proper trolley-platform, although this is more indicated for off-peak hours. This operation would add more complexity if the employee is reading some traditional shopping lists parallel, but the extra effort is negligible in case of TSP-optimized lists, since an optimized merged list with indications of the order number for each article could be easily produced.

4 Conclusions

In this work we propose an in-store operations optimization to increase the efficiency of in-store pickers in a click-and-collect service system relying on human workforce to collect articles from the shop. We discussed the benefit of reordering shopping lists according to the optimal TSP solution, and we provided some simulations in a real-world store. This illustrative example demonstrates the potential of the approach. Even more remarkable improvements could be achieved in stores with complex topology and pickers handling collection tasks in parallel. With online shopping getting more and more prominent, also due to the COVID-

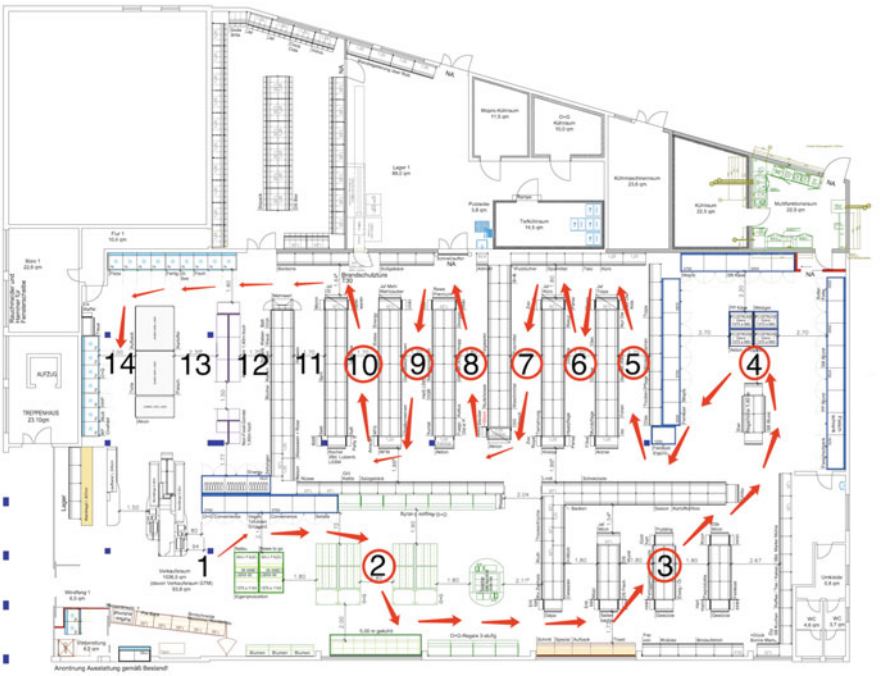


Fig. 3 Travelling path of a pick-up task with a fully optimized list according to the optimal TSP solution

19 pandemic, the potential time-save provided by the optimized solutions could definitely make the difference.

Acknowledgments Xiaochen Chou was supported by the Swiss National Science Foundation through grants 200020-182360: "Machine learning and sampling-based metaheuristics for stochastic vehicle routing problems".

References

1. Battini, D., Glock, C.H., Grosse, E.H., et al.: Human energy expenditure in order picking storage assignment: a bi-objective method. Comput. Ind. Eng. **94**, 147–157 (2016)
2. Bell, D.R., Gallino, S., Moreno, A.: How to win in an omni-channel world. MIT Sloan Manag. Rev. **56**, 45–55 (2014)
3. Bijmolt, T.H., Broekhuis, M., De Leeuw, S., et al.: Challenges at the marketing-operations interface in omni-channel retail environments. J. Bus. Res. **122**, 864–874 (2021)
4. Boysen, N., De Koster, R., Weidinger, F.: Warehousing in the e-commerce era. A survey. Eur. J. Oper. Res. **277**, 396–411 (2019)
5. Boysen, N., Füssler, D., Stephan, K.: See the light. Optimization of put-to-light order picking systems. Naval Res. Logist. **67**, 3–20 (2020)

6. Boysen, N., Koster, R., de, Füssler, D.: The forgotten sons. Warehousing systems for brick-and-mortar retail chains. Eur. J. Oper. Res. **288**, 361–381 (2021)
7. Calzavara, M., Glock, C.H., Grosse, E.H., et al.: An integrated storage assignment method for manual order picking warehouses considering cost, workload and posture. Int. J. Prod. Res. **57**, 2392–2408 (2019)
8. Cragg, T., Loske, D.: Perceived work autonomy in order picking systems. An empirical analysis. IFAC-PapersOnLine **52**, 1872–1877 (2019)
9. Dantzig, G., Fulkerson, R., Johnson, S.: Solution of a large-scale traveling-salesman problem. J. Oper. Res. Soc. Am. **2**(4), 393–410 (1954)
10. Gao, F., Su, X.: Omnichannel retail operations with buy-online-and-pick-up-in-store. Manag. Sci. **63**, 2478–2492 (2017)
11. Grand View Research. Online grocery market size, share and trends. Analysis report by product type, region, and segment forecasts for 2020–2027 (2020). https://www.grandviewresearch.com/industry-analysis/online-grocery-market
12. Grosse, E.H., Glock, C.H.: The effect of worker learning on manual order picking processes. Int. J. Prod. Econ. **170**, 882–890 (2015)
13. Hübner, A., Wollenburg, J., Holzapfel, A.: Retail logistics in the transition from multi-channel to omni-channel. Int. J. Phys. Distrib. Logist. Manag. **46**, 562–583 (2016)
14. Hui, S.K., Fader, P.S., Bradlow, E.T.: The traveling salesman goes shopping: the systematic deviations of grocery paths from TSP optimality. Market. Sci. **28**(3), 566–572 (2009)
15. Kim, K., Han, S.-L., Jang, Y.-Y., et al.: The effects of the antecedents of buy-online-pick-up-in-store service on consumer's BOPIS choice behaviour. Sustainability (Switzerland) **12**, 1–15 (2020)
16. Kong, R., Luo, L., Chen, L., et al.: The effects of BOPS implementation under different pricing strategies in omnichannel retailing. Transp. Res. E: Logist. Transp. Rev. **141**, 102014 (2020)
17. Lee, Y., Choi, S., Field, J.M.: Development and validation of the pick-up service quality scale of the buy-online-pick-up-in-store service. Oper. Manag. Res. **13**, 218–232 (2020)
18. Li, Z., Yang, W., Jin, H.S., et al.: Omnichannel retailing operations with coupon promotions. J. Retail. Consum. Serv. **58**, 102324 (2021)
19. Loske, D.: The impact of COVID-19 on transport volume and freight capacity dynamics. An empirical analysis in German food retail logistics. Transp. Res. Interdiscip. Perspect. **6**, 100165 (2020)
20. Matusiak, M., de Koster, R., Saarinen, J.: Utilizing individual picker skills to improve order batching in a warehouse. Eur. J. Oper. Res. **263**, 888–899 (2017)
21. Pardines, I., Lopez, V.: Shop&Go: TSP heuristics for an optimal shopping with smartphones. Sci. China Inform. Sci. **56**, 1–12 (2013)
22. Wang, Y., Xu, R., Schwartz, M., et al.: COVID-19 and retail grocery management: insights from a broad-based consumer survey. IEEE Eng. Manag. Rev. **48**, 202–211 (2020)
23. Weyland, D., Montemanni, R., Gambardella, L.M.: Heuristics for the probabilistic travelling salesman problem with deadlines based on quasi-parallel Monte Carlo sampling. Comput. Oper. Res. **40**(7), 1661–1670 (2013)
24. Wollenburg, J., Hübner, A., Kuhn, H., et al.: From bricks-and-mortar to bricks-and-clicks. Int. J. Phys. Distrib. Logist. Manag. **48**, 415–438 (2018)

An Optimization Model for the Evacuation Time in the Presence of Delay

Patrizia Daniele, Ornella Naselli, and Laura Scrimali

Abstract Natural disasters may have devastating effects on communities and affected areas. As a consequence, decision-makers have to be proactive and able to develop efficient rescue plans to save lives and prevent further damages. In this paper, we address the issue of planning the emergency evacuation of occupants of a building after a disaster event like a landslide. In particular, we propose a network model that minimizes both the travel time and the delay of evacuating. We also introduce a measure of the physical difficulties of evacuees and a parameter associated with the severity of the disaster. We then derive the variational inequality formulation. In order to illustrate the modeling framework, we present a numerical example.

Keywords Evacuation plans · Variational inequality · Lagrange duality · Utility

1 Introduction

Natural disasters related to ground movements, such as landslides, can be complicated and unpredictable and, therefore, difficult to risk assess. Landslides can occur in almost every country and can cause significant damage. Also, climate changes may increase the risk: more heavy rain and melting of local permafrost in some mountain areas and variations in ice temperature and local water level can increase the risk of a landslide. Landslides are one of the most relevant geomorphological hazards in a country, because of the high levels of people affected, destruction of assets and disruption of economic and social activities.

Italy is one of the European countries most affected by landslides, with 620,808 landslides in an area of 23,700 km^2, which is equal to 7.9% of the national territory (see Fig. 1). These data derive from the project of Inventario dei Fenomeni Franosi

P. Daniele (✉) · O. Naselli · L. Scrimali
Department of Mathematics and Computer Science, University of Catania, Catania, Italy
e-mail: patrizia.daniele@unict.it; naselli@dmi.unict.it; scrimali@dmi.unict.it

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_16

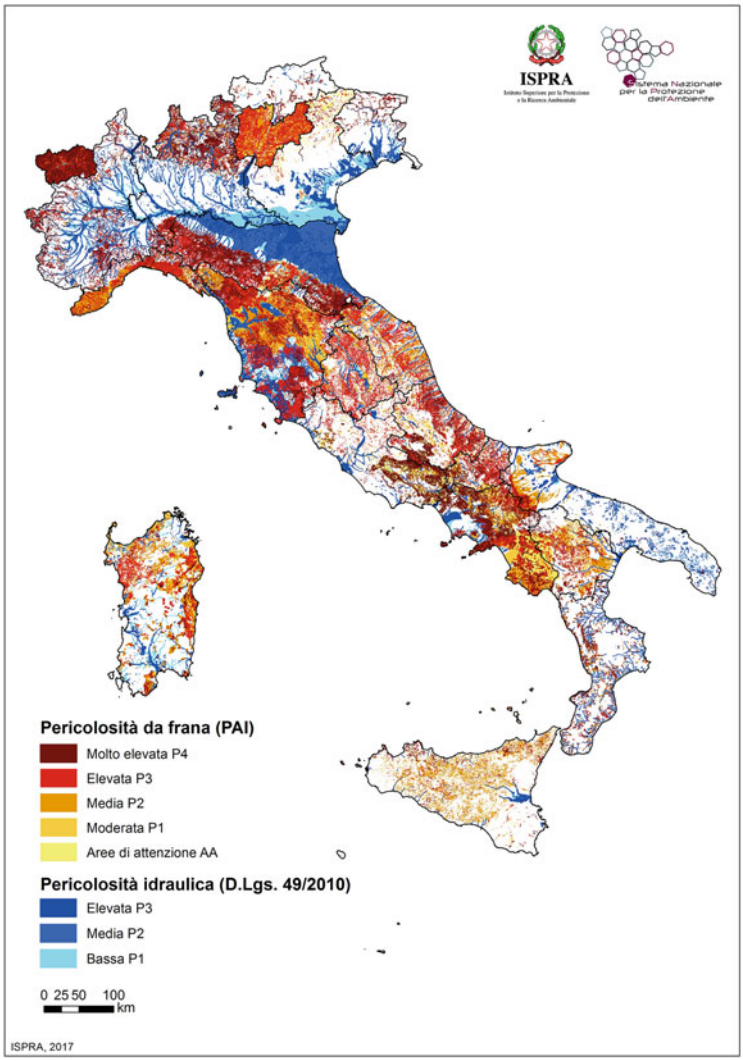


Fig. 1 Italian hydrogeological danger distribution

in Italy (IFFI Project) carried out by ISPRA (Superior Institute for Protection and Environmental Research) and the Regions and Autonomous Provinces, according to standardized and shared methods. About a third of the total landslides in Italy are rapid kinematic phenomena (collapses, rapid flows of mud and debris), characterized by high speeds, up to a few meters per second, and by high destructiveness, often with serious consequences in terms of loss of human lives. Other types of movements (e.g. slow flows, complex landslides), characterized by moderate or slow

speeds, can cause significant damage to residential areas and linear communication infrastructures.

The hydrogeological instability essentially includes two categories of events: landslides and floods. To get an idea of the size of the problem, we remember that since the beginning of the century there have been more than 4000 serious hydrogeological instability events that have caused great damage to people, houses and infrastructures, but, above all, they caused about 12,600 dead, missing and injured people and the number of missing people exceeds 700 thousand.

Almost 4% of Italian buildings (over 550 thousand) are located in areas with high and very high landslide danger and more than 9% (over 1 million) in flood areas. So, it is very important to be prepared and to reduce the total time for evacuation of a building in the case of a landslide or any other disaster.

Our aim is to propose an evacuation planning model that optimally assigns the shortest and safest paths, in order to minimize the total evacuation time and save the lives of the occupants. In particular, we propose a multicriteria evacuation model where the population at risk is evacuated, following criteria such as the total travel time and the total delay. We also introduce a measure of the physical difficulties of evacuees and a parameter associated with the severity of the disaster. This allows our model to be flexible and able to handle large-scale problems. In addition, it allows for the applications to different disaster scenarios. The optimization model that we develop is then formulated as a variational inequality (see [10, 12]), and an analysis of associated Lagrange multipliers is provided (see [3–5, 15]).

The problem of evacuation plans has been deeply studied in the literature.

In [6], the authors apply network flow techniques to find good exit selections for evacuees in an emergency evacuation and present two algorithms for computing exit distributions using both classical flows and flows over time which are well known from combinatorial optimization.

In [8], the authors present models and algorithms which can be applied to evacuation problems related to building evacuation, but which are applicable also to regional evacuation. For all the models time is the main parameter.

In [9], the authors present two different emergency evacuation models on the basis of the maximum flow model (MFM) and the minimum-cost maximum flow model (MC-MFM), and propose corresponding algorithms for the evacuation from one source node to one designated destination (one-to-one evacuation). Then, they extend the model from one source node to many designated destinations (one-to-many evacuation).

In [11], the authors propose an evacuation model which combines a heuristic algorithm and a network flow control, taking into account routes capacity constraints. They aim at minimizing the total evacuation time for all people.

In [17], a game-theoretical model to study cooperative and competitive behaviors of evacuating people during an emergency is proposed. The authors integrate a game-theoretical model with a cellular automation model of evacuation dynamics, and simulate the motions of crowds based on their competitive and cooperative strategies.

In this paper, for the first time, starting from a network model, we use the variational inequality formulation to obtain a characterization of the optimization problem consisting in minimizing the total evacuation time and, as far as we know, this methodology is innovative compared to the existing ones.

Such a methodology and the related computational procedures have been widely applied to solve real-world problems, such as static and dynamic traffic network equilibrium problems, spatial price equilibrium problems, oligopolistic market equilibrium problems, financial equilibrium problems, migration equilibrium problems, as well as environmental network and ecology problems, supply chain network equilibrium problems, cybersecurity networks, and even the Internet (see, for instance, [1, 2, 7, 12, 13, 15, 16] and the references therein.)

We also emphasize that variational inequality theory has revealed to be a powerful instrument in order to study complex decision-making behavior on networks, with the associated nodes, links, and induced flows. Therefore, characterizing our problem as a variational inequality, we may have recourse to all the well-established tools of the variational inequality theory, and ensure existence of solutions, qualitative analysis, and computational results.

The structure of this paper is as follows. In Sect. 2, we present the evacuation model and derive the variational inequality formulation. In Sect. 3, we provide a numerical example. Finally, we present our conclusions in Sect. 4.

2 The Mathematical Model

We consider a network as the one depicted in Fig. 1, where there is a building with I different rooms which are connected with J different stairs. Since different rooms are likely to share a part of their path towards the stairs as well as the existence of multiple floors leads to divide the stairs into pieces between the floors so that different levels of congestion on each piece are taken into account, we are considering a graph with transit nodes between rooms and stairs (the meeting points) and between stairs and exits (the lobbies). In turn, from the stairs it is possible to reach H different exit points. Normally, people will choose the closest stairs or exits, but, in case one of such points is particularly crowded or congested or blocked due to the disaster, then the evacuees can also choose alternative exits. The links between the first and the second level of nodes in the network represent all the possible connections between the rooms of the building and the stairs, as well as the links between the second and the third level of nodes in the network represent all the possible connections between the stairs and the final exits of the building (Fig. 2).

We denote by p_i^l the initial population in room A_i, $i = 1, \ldots, I$ of type l, $l = 1, \ldots, L$ and by $P = \sum_{l=1}^{L} \sum_{i=1}^{n} p_i^l$ the total population present in the building. Indeed, in our model we distinguish different types of individuals, in relation to their physical abilities. So, the apex l, $l = 1, \ldots, L$ represents the different types

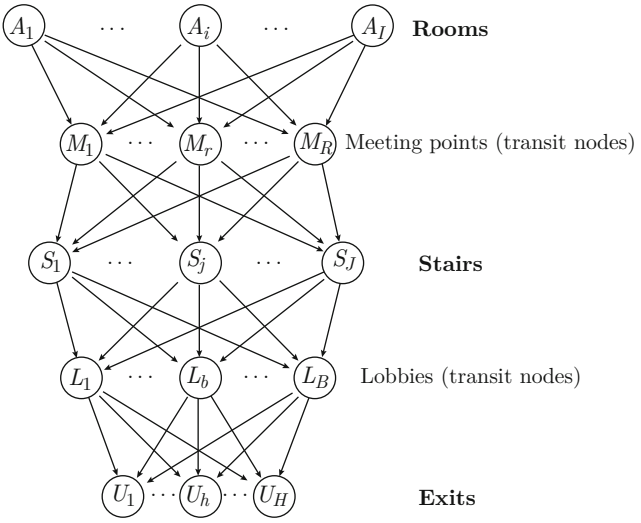


Fig. 2 Building network

of evacuated people. Moreover, let f_{ij}^l and g_{jh}^l be the flows of evacuees of type l in a time unit from A_i to S_j and from S_j to U_h, for $i = 1, \ldots, I$, $j = 1, \ldots, J$, and $h = 1, \ldots, H$, respectively. Since in a building the stairs are usually narrow spaces, we assume that u_j is the maximum allowed capacity in S_j, $j = 1, \ldots, J$. So, the following condition has to be satisfied:

$$\sum_{l=1}^{L} \sum_{i=1}^{I} \beta_{ij}^l f_{ij}^l \leq u_j, \quad \forall j = 1, \ldots, J, \tag{1}$$

where β_{ij}^l indicates the portion of people of type l that decide to evacuate from room A_i using the stair S_j. Further, we denote by t_{ij}^l the travel time spent by a person of type l to go from A_i to S_j through one of the meeting points M_r, $r = 1, \ldots, R$ and we assume it is a function of the flow of people from A_i to S_j:

$$t_{ij}^l = t_{ij}^l(f_{ij}^l), \quad i = 1, \ldots, I, \ j = 1, \ldots, J, \ l = 1, \ldots, L.$$

Analogously, we denote by τ_{jh}^l the travel time spent by a person of type l to go from S_j to U_h through one of the lobbies L_b, $b = 1, \ldots, B$ and we assume it is a function of the flow of people from S_j to U_h:

$$\tau_{jh}^l = \tau_{jh}^l(g_{jh}^l), \quad j = 1, \ldots, J, \ h = 1, \ldots, H, \ l = 1, \ldots, L.$$

Now, we introduce the delay functions which involve time, associated with the links from A_i to S_j and from S_j to U_h, respectively, and we assume they depend on the

Table 1 Functions and parameters

Symbols	Definitions
$A = \{A_i : i = 1, \ldots, I\}$	Set of rooms
$M = \{M_r : r = 1, \ldots, R\}$	Set of meeting points
$S = \{S_j : j = 1, \ldots, J\}$	Set of stairs
$L = \{L_b : b = 1, \ldots, B\}$	Set of lobbies
$U = \{U_h : h = 1, \ldots, H\}$	Set of exits
$E = \{l : l = 1, \ldots, L\}$	Set of types of people to be evacuated
p_i^l	Population of type l in node A_i
$P = \sum_{l=1}^{L} \sum_{i=1}^{I} p_i^l$	Population of any type to be evacuated
u_j	Maximum capacity of stair S_j
β_{ij}^l	Portion of people of type l evacuating from A_i through S_j
f_{ij}^l	Flow of people of type l on the link from A_i to S_j
g_{jh}^l	Flow of people of type l on the link from S_j to U_h
$t_{ij}^l(f_{ij}^l)$	Travel time on the link from A_i to S_j for a person of type l
$\tau_{jh}^l(g_{jh}^l)$	Travel time on the link from S_j to U_h for a person of type l
$R_{ij}^{1l}(f_{ij}^l)$	Delay function on the link from A_i to S_j for a person of type l
$R_{jh}^{2l}(g_{jh}^l)$	Delay function on the link from S_j to U_h for a person of type l
$\alpha^l \in [0, 1]$	Index measuring the physical difficulties of type l
$\sigma \in [0, 1]$	Severity coefficient of the disaster

flows on the links, namely:

$$R_{ij}^{1l} = R_{ij}^{1l}(f_{ij}^l) \text{ and } R_{jh}^{2l} = R_{jh}^{2l}(g_{jh}^l), \quad i = 1, \ldots, I,$$
$$j = 1, \ldots, J, \ h = 1, \ldots, H, \ l = 1, \ldots, L.$$

In addition, we consider two coefficients $\alpha^l, \sigma \in [0, 1]$ representing the measure of the physical difficulties for an evacuee of type l and the severity of the disaster, respectively.

We group all the functions and parameters in Table 1.

The purpose of our model is to minimize the total evacuation time, denoted by $ET(f, g)$, given by the sum of the total travel times and the total delay. Hence, we are interested in solving the following optimization problem:

$$\min ET(f, g) = \min \left\{ \sum_{l=1}^{L} \left[\sum_{i=1}^{I} \sum_{j=1}^{J} t_{ij}^l(f_{ij}^l) f_{ij}^l + \left(1 + \alpha^l\right) \sum_{j=1}^{J} \sum_{h=1}^{H} \tau_{jh}^l(g_{jh}^l) g_{jh}^l \right.\right.$$
$$\left.\left. + \sigma \sum_{i=1}^{I} \sum_{j=1}^{J} R_{ij}^{1l}(f_{ij}^l) f_{ij}^l + \sigma \sum_{j=1}^{J} \sum_{h=1}^{H} R_{jh}^{2l}(g_{jh}^l) g_{jh}^l \right] \right\} \tag{2}$$

under (1) and the following constraints:

$$\sum_{l=1}^{L}\sum_{i=1}^{n} f_{ij}^{l} \geq \sum_{l=1}^{L}\sum_{h=1}^{k} g_{jh}^{l},\ \forall j = 1, \ldots, J; \tag{3}$$

$$\sum_{j=1}^{J} f_{ij}^{l} \leq p_i^{l}, \quad \forall i,\ \forall l; \tag{4}$$

$$\sum_{j=1}^{J} f_{ij}^{l} \geq .5 p_i^{l}, \quad \forall i,\ \forall l; \tag{5}$$

$$\sum_{j=1}^{J}\sum_{h=1}^{H} g_{jh}^{l} \geq .5 p_i^{l}, \quad \forall l; \tag{6}$$

$$f_{ij}^{l} \geq 0, \quad \forall i,\ \forall j,\ \forall l; \quad g_{jh}^{l} \geq 0, \quad \forall j,\ \forall h,\ \forall l. \tag{7}$$

Constraint (3) states that, for every index j, the sum of the flows of people from any room A_i to S_j exceeds the sum of the flows of people of all types l from S_j to any exit U_h. Constraint (4) establishes that people moving on all the links cannot exceed the total population on the building. With constraints (5) and (6) we guarantee that at least 50% of persons evacue from every room and from the building, respectively. Finally, constraints (7) are the nonengativity conditions of the flows.

Let us define the set of constraints as the feasible set $\mathbb{K}$ given by:

$$\mathbb{K} = \Bigg\{ (f, g) \in \mathbb{R}^{IJL+JHL} :\ f_{ij}^{l} \geq 0,\ \forall i,\ \forall j,\ \forall l;\ g_{jh}^{l} \geq 0,\ \forall j,\ \forall h,\ \forall l;$$
$$\sum_{l=1}^{L}\sum_{i=1}^{I} \beta_{ij}^{l} f_{ij}^{l} - u_j \leq 0,\ \forall j;\ \sum_{l=1}^{L}\sum_{h=1}^{k} g_{jh}^{l} - \sum_{l=1}^{L}\sum_{i=1}^{I} f_{ij}^{l} \leq 0,\ \forall j;$$
$$\sum_{j=1}^{J} f_{ij}^{l} - p_i^{l} \leq 0,\ \forall i,\ \forall l;\ .5 p_i^{l} - \sum_{j=1}^{J} f_{ij}^{l} \leq 0,\ \forall i,\ \forall l;\ .5 p_i^{l} - \sum_{j=1}^{J}\sum_{h=1}^{H} g_{jh}^{l} \leq 0,\ \forall l \Bigg\}$$

and assume that the travel time and delay functions multiplied by the respective flows are continuously differentiable and convex. Then, since the set $\mathbb{K}$ is closed, bounded, and convex, applying the classical theory on the variational inequalities

(see, for instance, [10] or [12]), problem can be characterized by means of the following variational inequality:

Find $(f^*, g^*) \in \mathbb{K}$ such that:

$$\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{l=1}^{L}\left[\frac{\partial t_{ij}^{l}(f_{ij}^{l*})}{\partial f_{ij}^{l}} f_{ij}^{l*} + t_{ij}^{l}(f_{ij}^{l*}) + \sigma\left(\frac{\partial R_{ij}^{1l}(f_{ij}^{l*})}{\partial f_{ij}^{l}} f_{ij}^{l*} + R_{ij}^{1l}(f_{ij}^{l*})\right)\right] \times \left(f_{ij}^{l} - f_{ij}^{l*}\right)$$
$$+\sum_{j=1}^{J}\sum_{h=1}^{H}\sum_{l=1}^{L}\left[(1+\alpha^{l})\left(\frac{\partial \tau_{jh}^{l}(g_{jh}^{l*})}{\partial g_{jh}^{l}} g_{jh}^{l*} + \tau_{jh}^{l}(g_{jh}^{l*})\right)\right. \tag{8}$$
$$\left. + \sigma\left(\frac{\partial R_{jh}^{2l}(g_{jh}^{*})}{\partial g_{jh}} g_{jh}^{l*} + R_{jh}^{2l}(g_{jh}^{l*})\right)\right] \times \left(g_{jh}^{l} - g_{jh}^{l*}\right) \geq 0, \quad \forall (f, g) \in \mathbb{K}.$$

Now, taking into account the Lagrange multipliers associated with the constraints defining the feasible set $\mathbb{K}$, and using the same technique as in [1, 2, 13, 16], we obtain an important result.

We can consider the following Lagrange function:

$$\mathscr{L}(f, g, \gamma, \delta, \eta, \vartheta, \lambda, \mu, \nu) = V(f, g) + \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{l=1}^{L}\gamma_{ij}^{l}(-f_{ij}^{l}) + \sum_{j=1}^{J}\sum_{h=1}^{H}\sum_{l=1}^{L}\delta_{jh}^{l}(-g_{jh}^{l})$$
$$+\sum_{j=1}^{J}\eta_{j}\left(\sum_{i=1}^{I}\sum_{l=1}^{L}\beta_{ij}^{l} f_{ij}^{l} - u_{j}\right) + \sum_{j=1}^{J}\vartheta_{j}\left(\sum_{l=1}^{L}\sum_{h=1}^{H} g_{jh}^{l} - \sum_{l=1}^{L}\sum_{i=1}^{I} f_{ij}^{l}\right)$$
$$+\sum_{l=1}^{L}\sum_{i=1}^{I}\lambda_{i}^{l}\left(\sum_{j=1}^{J} f_{ij}^{l} - p_{i}^{l}\right) + \sum_{l=1}^{L}\sum_{i=1}^{n}\mu_{i}^{l}\left(.5p_{i}^{l} - \sum_{j=1}^{J} f_{ij}^{l}\right) + \sum_{l=1}^{L}\nu^{l}\left(.5p_{i}^{l} - \sum_{j=1}^{J}\sum_{h=1}^{H} g_{jh}^{l}\right)$$

where $V(f, g)$ is the left-hand side of (8) and $f \in \mathbb{R}^{IJL}$, $g \in \mathbb{R}^{JHL}$, $\gamma \in \mathbb{R}_{+}^{IJL}$, $\delta \in \mathbb{R}_{+}^{JHL}$, $\eta \in \mathbb{R}_{+}^{J}$, $\vartheta \in \mathbb{R}_{+}^{J}$, $\lambda \in \mathbb{R}_{+}^{IL}$, $\mu \in \mathbb{R}_{+}^{IL}$, $\nu \in \mathbb{R}_{+}^{L}$.

Then, the following result holds true.

Theorem 1 *If $(f^*, g^*) \in \mathbb{K}$ is a solution to variational inequality (8), then the Lagrange multipliers $\bar{\gamma} \in \mathbb{R}_{+}^{IJL}$, $\bar{\delta} \in \mathbb{R}_{+}^{JHL}$, $\bar{\eta} \in \mathbb{R}_{+}^{J}$, $\bar{\vartheta} \in \mathbb{R}_{+}^{J}$, $\bar{\lambda} \in \mathbb{R}_{+}^{IL}$, $\bar{\mu} \in \mathbb{R}_{+}^{IL}$, and $\bar{\nu} \in \mathbb{R}_{+}^{L}$ do exist, and for all i, j, h, and l, the following conditions hold true:*

$$\bar{\gamma}_{ij}^{l}(-f_{ij}^{*}) = 0, \quad \bar{\delta}_{jh}^{l}(-g_{jh}^{*}) = 0,$$
$$\bar{\eta}_{j}\left(\sum_{i=1}^{I}\sum_{l=1}^{L}\beta_{ij}^{l} f_{ij}^{l*} - u_{j}\right) = 0, \quad \bar{\vartheta}_{j}\left(\sum_{l=1}^{L}\sum_{h=1}^{H} g_{jh}^{l*} - \sum_{l=1}^{L}\sum_{i=1}^{I} f_{ij}^{l*}\right) = 0,$$
$$\bar{\lambda}_{i}^{l}\left(\sum_{j=1}^{J} f_{ij}^{l*} - p_{i}^{l}\right) = 0, \ \bar{\mu}_{i}^{l}\left(.5p_{i}^{l} - \sum_{j=1}^{J} f_{ij}^{*}\right) = 0, \ \bar{\nu}^{l}\left(.5p_{i}^{l} - \sum_{j=1}^{J}\sum_{h=1}^{H} g_{jh}^{l*}\right) = 0,$$

$$\frac{\partial t_{ij}^{l}(f_{ij}^{l*})}{\partial f_{ij}^{l}} f_{ij}^{l*} + t_{ij}^{l}(f_{ij}^{l*}) + \sigma \left(\frac{\partial R_{ij}^{1l}(f_{ij}^{l*})}{\partial f_{ij}^{l}} f_{ij}^{l*} + R_{ij}^{1l}(f_{ij}^{l*}) \right)$$
$$-\bar{\gamma}_{ij}^{l} + \bar{\eta}_{j}\beta_{ij}^{l} - \bar{\vartheta}_{j} + \bar{\lambda}_{i}^{l} - \bar{\mu}_{i}^{l} = 0,$$

$$(1+\alpha^{l}) \left(\frac{\partial \tau_{jh}^{l}(g_{jh}^{l*})}{\partial g_{jh}^{l}} g_{jh}^{l*} + \tau_{jh}^{l}(g_{jh}^{l*}) \right) + \sigma \left(\frac{\partial R_{jh}^{2l}(g_{jh}^{*})}{\partial g_{jh}} g_{jh}^{l*} + R_{jh}^{2l}(g_{jh}^{l*}) \right)$$
$$-\bar{\delta}_{jh}^{l} + \bar{\vartheta}_{j} - \bar{\nu}^{l} = 0.$$

Moreover, the strong duality also holds true; namely:

$$V(f^{*}, g^{*}) = \min_{\mathbb{K}} V(f, g) = \max_{(\gamma,\delta,\eta,\vartheta,\lambda,\mu,\nu)} \inf_{(f,g)} \mathscr{L}(f, g, \gamma, \delta, \eta, \vartheta, \lambda, \mu, \nu).$$

3 Numerical Illustration

In order to validate our model, we now provide a small numerical example.

We consider a public building with a street-level floor and two floors above. We assume that 100 persons are located in the second floor of the building and are distributed in three different rooms. A landslide impacts the area of the building, so that people have to evacuate, choosing one of the two existing stairs that leads to three possible exits. We also suppose that there are two types of people, according to their physical difficulties. The parameter values are:

$$(p_{i}^{1})_{i=1,\ldots,3} = (10, 20, 20), (p_{i}^{2})_{i=1,\ldots,3} = (15, 15, 20),$$
$$\alpha_{1} = 0,\ \alpha_{2} = 0.3,\ \sigma = 0.5,\ u_{1} = 15,\ u_{2} = 15,\ \beta_{ij}^{l} = 0.35,\ \forall i, j, l.$$

The total travel time and the delay functions are reported in Tables 2 and 3.

We solved the resulting variational inequality applying the extragradient method with constant step length as in [14] (see also [7]), implemented as M-script files of

Table 2 Travel times and delay functions for occupants of type 1

$t_{ij}^{1}(f_{ij})$	$\tau_{jh}^{1}(g_{jh})$	$R_{ij}^{11}(f_{ij})$	$R_{jh}^{21}(g_{jh})$
$2f_{11}^{2} + f_{11}$	$2g_{11}^{2} + 25g_{11}$	$4f_{11}^{2} + 15f_{11}$	$2g_{11}^{2} + 25g_{11}$
$0.5f_{12}^{2} + 4f_{12}$	$g_{12}^{2} + 5g_{12}$	$f_{12}^{2} + 4f_{12}$	$g_{12}^{2} + 5g_{12}$
$f_{21}^{2} + 4f_{21}$	$5g_{13}^{2} + 50g_{13}$	$f_{21}^{2} + 4f_{21}$	$5g_{13}^{2} + 50g_{13}$
$f_{22}^{2} + 3f_{22}$	$g_{21}^{2} + 2g_{21}$	$f_{22}^{2} + 3f_{22}$	$g_{21}^{2} + 2g_{21}$
$f_{31}^{2} + 15f_{31}$	$g_{22}^{2} + g_{22}$	$f_{31}^{2} + 15f_{31}$	$g_{22}^{2} + g_{22}$
$f_{32}^{2} + 30f_{32}$	$g_{23}^{2} + 5g_{23}$	$f_{32}^{2} + 30f_{32}$	$g_{23}^{2} + 5g_{23}$

Table 3 Travel times and delay functions for occupants of type 2

$t^2_{ij}(f_{ij})$	$\tau^2_{jh}(g_{jh})$	$R^{12}_{ij}(f_{ij})$	$R^{22}_{jh}(g_{jh})$
$4f^2_{11} + 2f_{11}$	$3g^2_{11} + 15g_{11}$	$2f^2_{11} + 25f_{11}$	$3g^2_{11} + 15g_{11}$
$f^2_{12} + 5f_{12}$	$2g^2_{12} + 50g_{12}$	$4f^2_{12} + 40f_{12}$	$2g^2_{12} + 50g_{12}$
$1.5f^2_{21} + 12f_{21}$	$8g^2_{13} + 5g_{13}$	$4f^2_{21} + 20f_{21}$	$8g^2_{13} + 5g_{13}$
$2f^2_{22} + 5f_{22}$	$2g^2_{21} + 10g_{21}$	$6f^2_{22} + 5f_{22}$	$2g^2_{21} + 10g_{21}$
$1.5f^2_{31} + 20f_{31}$	$g^2_{22} + 15g_{22}$	$2f^2_{31} + 32f_{31}$	$g^2_{22} + 15g_{22}$
$2f^2_{32} + 30f_{32}$	$g^2_{23} + 50g_{23}$	$3f^2_{32} + 22f_{32}$	$g^2_{23} + 50g_{23}$

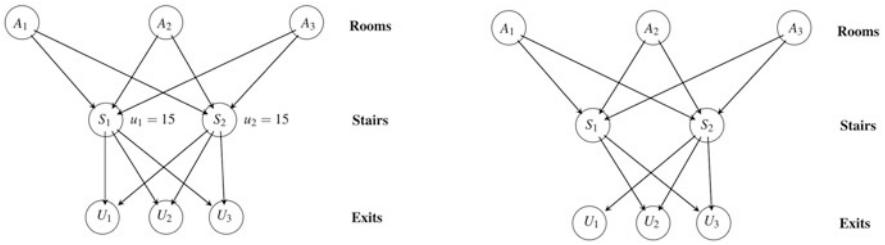


Fig. 3 Network topology and evacuation paths of the example

Table 4 Optimal flows on the paths used for evacuation

Flows	Optimal values
(f^1_{11}, f^2_{11})	(2.4927;0)
(f^1_{12}, f^2_{12})	(5.0037;5.0037)
(f^1_{22}, f^2_{22})	(10.0055;0)
(f^1_{31}, f^2_{31})	(5.8297;5.8297)
(f^1_{32}, f^2_{32})	(8.3407;0)
(g^1_{12}, g^2_{12})	(4.5044;8.1333)
(g^1_{13}, g^2_{13})	(4.5044;5.6222)
(g^1_{21}, g^2_{21})	(1.9934;5.6222)
(g^1_{22}, g^2_{22})	(9.4934;5.6222)
(g^1_{23}, g^2_{23})	(4.5044;0)

MatLab. We note that our problem satisfies the assumptions needed to ensure the existence of solutions as well as the convergence of the algorithm.

In Fig. 3, we represent the network topology of the building on the left, and the optimal path distribution on the right. The optimal evacuation flows are given in Table 4. The total evacuation time, namely the value of the objective function $ET(f^*, g^*)$ (see objective function (2)) is 8.3833 h. This value takes into account that displacements and ground movements, due to the landslide, may cause structural damages to the building (extensive cracks, distorsions in pillars and columns, tilting of floors and walls, obstructed doors, etc.). This makes the evacuation time increase. Finally, we note that all the people in the building are able to evacuate.

4 Conclusions

In this paper, we introduced an evacuation planning model that identifies the optimal flows of people who must be evacuated from a building after a landslide. The multicriteria objective of the problem was to minimize both the total travel time and the total delay, which were influenced by the physical difficulties of evacuees and the severity of the disaster. We then proposed a variational inequality formulation of the model and provided its dual problem. In addition, we showed an alternative formulation based on the Lagrange multipliers associated with the constraints. They may have a crucial role in order to capture and predict the variation in the escape speed. Finally, we provided a numerical example that emphasized how the model developed in this paper can be used by policy-makers to plan emergency evacuation after a natural disaster.

Future research may include extending this framework to assess sinergies among individuals who could act as a group/coalition.

The results in this paper add to the growing literature of operations research for management of evacuation plans.

Acknowledgments The research was partially supported by the research projects PON SCN 00451 CLARA—CLoud plAtform and smart underground imaging for natural Risk Assessment, Smart Cities and Communities and Social Innovation, and "Programma ricerca di ateneo UNICT 2020–2022 linea 2-OMNIA" of Catania. These support are gratefully acknowledged.

References

1. Caruso, V., Daniele, P.: A network model for minimizing the total organ transplant costs. Eur. J. Oper. Res. **266**, 652–662 (2018)
2. Colajanni, G., Daniele, P., Giuffrè, S., Nagurney, A.: Cybersecurity investments with nonlinear budget constraints and conservation laws: variational equilibrium, marginal expected utilities, and Lagrange multipliers. Int. Trans. Oper. Res. **25**, 1443–1464 (2018)
3. Daniele, P., Giuffrè, S., Idone, G., Maugeri, A.: Infinite dimensional duality and applications. Math. Ann. **339**, 221–239 (2007)
4. Daniele, P., Giuffrè, S., Lorino, M.: Functional inequalities, regularity and computation of the deficit and surplus variables in the financial equilibrium problem. J. Global Optim. **65**(1), 575–596 (2015)
5. Daniele, P., Giuffrè, S., Maugeri, A., Raciti, F.: Duality theory and applications to unilateral problems. J. Optim. Theory Appl. **162**, 718–734 (2014)
6. Dressler, D., Gross, M., Kappmeier, J-P., Kelter, T., Kulbatzki, J., Plümpe, D., Schlechter, G., Schmidt, M., Skutella, M., Temme, S.: On the use of network flow techniques for assigning evacuees to exits, in *International Conference on Evacuation Modeling and Management, Procedia Engineering*, vol. 3 (2010), pp. 205–215
7. Facchinei, F., Pang, J.S.: *Finite-Dimensional Variational Inequalities and Complementarity Problems*, vol. I (Springer, New York, 2003)
8. Hamacher, H.W., Tjandra, S.A.: Mathematical Modelling of evacuation problems: a state of the art. Berichte des Fraunhofer ITWM, Nr. 24 (2001). https://kluedo.ub.uni-kl.de/frontdoor/deliver/index/docId/1477/file/bericht24.pdf

9. Li, G., Zhang, L., Wang, Z.: Optimization and planning of emergency evacuation routes considering traffic control. Sci. World J. **2014**, 164031. https://doi.org/10.1155/2014/164031
10. Kinderlehrer, D., Stampacchia, G.: *An Introduction to Variational Inequalities and Their Applications* (Academic Press, New York, 1980)
11. Liu, C., Mao, Z., Fu, Z.: Emergency evacuation model and algorithm in the building with several exits. Proc. Eng. **135**, 12–18 (2016)
12. Nagurney, A.: *Network Economics: A Variational Inequality Approach*, 2nd edn. (revised) (Kluwer Academic Publishers, Boston, 1999)
13. Nagurney, A., Salarpour, M., Daniele, P.: An integrated financial and logistical game theory model for humanitarian organizations with purchasing costs, multiple freight service providers, and budget, capacity, and demand constraints. Int. J. Prod. Econ. **212**, 212–226 (2019)
14. Korpelevich, G.M.: The extragradient method for finding saddle points and other problems. Matekon **13**, 35–49 (1977)
15. Scrimali, L.: On the stability of coalitions in supply chain networks via generalized complementarity conditions. Netw. Spat Econ. (2019). https://doi.org/10.1007/s11067-019-09461-w
16. Toyasaki, F., Daniele, P., Wakolbinger, T.: A variational inequality formulation of equilibrium models for end-of-life products with nonlinear constraints. Eur. J. Oper. Res. **236**, 340–350 (2014)
17. Zheng, X., Cheng, Y.: Modeling cooperative and competitive behaviors in emergency evacuation: a game-theoretical approach. Comput. Math. Appl. **62**, 4627–4634 (2011)

Additive Bounds for the Double Traveling Salesman Problem with Multiple Stacks

Luca Diedolo and Giovanni Righini

Abstract The Double TSP with Multiple Stacks is a challenging combinatorial optimization problem, asking for two Hamiltonian cycles on two weighted graphs, a pick-up graph and a delivery graph; the two cycles originate from two given depots, one for each graph, and they visit the vertices in an order that allows a single vehicle to collect the pick-up items in a given number of stacks and to deliver them according to a Last-In-First-Out policy for each stack. We investigate the use of the additive bounding procedure, starting from the Held-Karp lower bound, within a branch-and-bound algorithm. Computational results show that this method often provides tighter bounds than the Double TSP relaxation.

Keywords Traveling Salesman Problem · Held-Karp lower bound · Additive bounding

1 The DTSPMS

The Double Traveling Salesman Problem with Multiple Stacks (DTSPMS) was been first described by Petersen et al. [9] as a variant of the Pickup and Delivery Traveling Salesman Problem (PDTSP).

The PDTSP is the problem where a single vehicle with potentially infinite capacity must serve a set of pickup and delivery requests. Each request requires to visit a given pickup vertex and a given delivery vertex in the graph. The objective is to find the minimum cost Hamiltonian cycle, taking into account that the pickup node must be visited before the delivery node for each pair.

L. Diedolo
ACT-OR, Verona, Italy
e-mail: luca.diedolo@ahead-research.com

G. Righini (✉)
Department of Computer Science, University of Milan, Milan, Italy
e-mail: giovanni.righini@unimi.it

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_17

In the DTSPMS, instead, pickup vertices and delivery vertices must be visited along two disjoint Hamiltonian cycles, each one starting and ending at a depot. Therefore we are given a pickup graph and a delivery graph of the same size; for each request the vehicle must visit a given pickup vertex in the pickup graph and a given delivery vertex in the delivery graph. In the remainder we assign an index in $\{1, \ldots, n\}$ to each request and to its corresponding pickup and delivery vertices.

Moreover, the vehicle loading compartment is divided into a specified number of fixed height stacks, managed according to a Last-In-First-Out policy: while visiting the pickup graph, items must be put on top of one of the stacks and while visiting the delivery graph items can be delivered only when they are on top of their stack.

Besides heuristic approaches, such as those of Felipe et al. [6], Chagas et al. [4] and Urrutia et al. [10], some exact algorithms have been developed, such as a branch-and-bound by Carrabs et al. [2] for the problem with 2 stacks. Currently the best known algorithm for computing exact solutions of the DTSPMS is a branch-and-cut algorithm developed by Alba Martinez et al. [1], that could solve some instances with up to 28 vertex pairs.

In this paper, as a first step to evaluate the strength of a new lower bounding technique based on additive bounds, we consider a slightly relaxed version of the problem, where the stacks have no fixed height, i.e. no capacity restrictions. The same uncapacitated version was also investigated by Casazza et al. [3] to shed light on some useful properties of the formulation.

The goal of our study is to investigate the potential usefulness and possible limitations of additive bounds, proposed by Fischetti and Toth [7]. The additive bounding technique is based on the availability of different lower bounding procedures for a combinatorial optimization problem. Instead of running each of them and then take the tightest lower bound obtained, one can sort the procedures and run each one on the graphs where the weights are the reduced costs produced by the previous one. The lower bounds computed by each procedure can be added up to produce a final lower bound.

This idea can be exploited within a branch-and-bound algorithm to solve the DTSPMS because the constraint set allows for different relaxations: in particular, we initially compute a relaxation of the Hamiltonian cycle constraints on both graphs, using the well-known Held and Karp method [8], so that reduced costs are also provided. Then, we compute minimum repair costs for the incompatibilities that are detected in the solution. This is done by computing a number of min cost shortest paths.

2 Lower Bounding the DTSPMS

The additive bounding procedure, to be executed for each sub-problem in a branch-and-bound tree, runs in three steps. In Step 1 (feasibility check) the two optimal Hamiltonian cycles on the two graphs are computed; an incompatibility graph is defined, to detect violations of the LIFO constraints. If no violations are detected,

then the sub-problem is solved to optimality. Otherwise, steps 2 and 3 are executed. In Step 2 (routing lower bound) a minimum cost 1-tree is computed for each graph and the corresponding reduced costs of the edges are recorded. In Step 3 (repair step) the minimum cost for repairing infeasibilities detected in Step 1 is computed, providing a final valid lower bound.

2.1 Step 1: Feasibility Check

Given the restrictions due to previous branching decisions (some edges in either graphs can be forbidden or fixed), the state-of-the-art TSP solver Concorde [5] is run to compute the optimal Hamiltonian cycles on the two graphs. Further details on how fixed and forbidden arcs are managed in this step are described in the subsection devoted to branching.

Since the graphs in DTSPMS instances are small, the computing time taken by Concorde is almost negligible. Then, the two solutions produced are analyzed. Consider an arbitrary orientation of the two Hamiltonian cycles; consider two distinct requests i and j. If vertices i and j appear in the same order along both cycles, then their requests must be assigned to different stacks, owing to the LIFO constraint. On the contrary, if they appear in a different order, then their requests must be assigned to different stacks if the orientation of one of the two cycles is reversed.

Hence, we define an incompatibility graph $\mathcal{G} = (\mathcal{V}, \mathcal{S}, \mathcal{R})$ with a vertex in $\mathcal{V}$ for each request and two disjoint edge sets $\mathcal{S}$ and $\mathcal{R}$: there is a *straight* edge $[i, j] \in \mathcal{S}$ if and only if i and j appear in the same order in both cycles and a *reverse* edge $[i, j] \in \mathcal{R}$ if and only if they appear in a different order. The incompatibility graph is therefore complete and all its edges are either in $\mathcal{S}$ or in $\mathcal{R}$.

Since s stacks are available on the vehicle, if $\mathcal{G}$ contains at least one clique of straight edges (st-clique) and at least one clique of reverse edges (rv-clique) of cardinality larger than s, then there is no orientation complying with the LIFO constraints.

2.2 Step 2: Routing Lower Bound

The aim of this step is to produce tight lower bounds to the Double TSP (disregarding the LIFO constraints) with the additional information provided by reduced costs. It is possible to do this efficiently owing to the well-known method due to Held and Karp [8]. For each graph a minimum cost 1-tree is computed, spanning all vertices with the exception of an arbitrarily selected vertex, indicated by d; then, two minimum cost edges incident to d are also selected, forming a minimum cost

spanning 1-tree. Edges incident to vertices with degree larger than two are penalized, by increasing their cost by Lagrangean multipliers in a subgradient optimization algorithm, illustrated in Algorithm 1, and the search for a minimum cost spanning 1-tree is iterated.

Algorithm 1 The subgradient optimization algorithm. Input: a weighted graph $(\mathcal{V}, \mathcal{E})$ with a cost function $c : \mathcal{E} \mapsto \mathbb{Z}$ and a small tolerance ϵ. Output: the best valid lower bound found within $MaxIter$ iterations or within $MaxIterNoImpr$ iterations without improvement with a step parameter smaller than ϵ

$MaxIter \leftarrow n^2$
$MaxIterNoImpr \leftarrow n$
$\lambda \leftarrow 2$
$LB \leftarrow 0$
$LastImpr \leftarrow 0$
$m \leftarrow 1$
$\pi^{(1)} \leftarrow 0$
$Stop \leftarrow false$
repeat
  for $[i, j] \in \mathcal{E}$ **do**
    $\overline{c}_{ij}^{(m)} \leftarrow c_{ij} + \pi_i^{(m)} + \pi_j^{(m)}$
  end for
  $T^{(m)} \leftarrow MinCost1tree(\mathcal{V}, \mathcal{E}, \overline{c}^{(m)})$
  $z^{(m)} = \overline{c}(T^{(m)}) - 2\sum_{i \in \mathcal{V}} \pi_i^{(m)}$
  if $z^{(m)} = UB$ **then**
    $Stop \leftarrow true$
  else
    if $z^{(m)} > LB$ **then**
      $LastImpr \leftarrow m$
      $LB \leftarrow z^{(m)}$
    end if
    if $m - LastImpr = MaxIterNoImpr$ **then**
      $\lambda \leftarrow \lambda/2$
      $LastImpr \leftarrow m$
      if $\lambda < \epsilon$ **then**
        $Stop \leftarrow true$
      end if
    end if
    $t^{(m)} \leftarrow \frac{\lambda(UB - z^{(m)})}{\sum_{i \in \mathcal{V}}(degree(i, T^{(m)}) - 2)^2}$
    for $i \in \mathcal{V}$ **do**
      $\pi_i^{(m+1)} \leftarrow \pi_i^{(m)} + t^{(m)}(degree(i, T^{(m)}) - 2)$
    end for
    $m \leftarrow m + 1$
  end if
until $Stop \vee (m = MaxIter)$
return LB

The sum of the costs of the two 1-trees is a valid lower bound for the DTSPMS, and it is indicated by LB_1.

Reduced costs are separately computed for edges incident to d and for edges in the minimum cost spanning tree:

- the reduced cost of each non-selected edge incident to d is the difference between its original cost and the cost of the second cheapest selected edge incident to d;
- the reduced cost $\overline{c}_{ij}$ of any other non-selected edge $[i, j]$ is the difference between its original cost c_{ij} and the maximum cost among the edges in the unique cycle formed by edge $[i, j]$ and the minimum cost spanning tree.

2.3 Step 3: Repair

In the third and final step, incompatibilities are considered and the minimum cost to repair each of them is computed. The aim of this step is to compute a lower bound to the cost to be paid for repairing each clique pair made by an st-clique and an rv-clique. In this step the edge weights on the graphs are the reduced costs $\overline{c}$.

There are two ways to repair an incompatibility: the first one, called *destructive* way, consists of destroying one of the two cliques by visiting its vertices in a different order; the second one, called *non-destructive* way, consists of reversing the order of the vertices of one of the two cliques with respect to the other. Both ways must be evaluated in order to retrieve the final repair cost.

Example Assume $s = 2$ and the two cycles, arbitrarily oriented, contain the sequences $[1, 2, 3]$ and $[4, 5, 6]$ in one graph and $[1, 2, 3]$ and $[6, 5, 4]$ in the other graph. Then $[1, 2, 3]$ is a straight clique of cardinality $s + 1$ and $[4, 5, 6]$ is a reverse clique of cardinality $s + 1$. The existence of this pair of cliques implies a violation of the LIFO constraints. The violation can be repaired in several ways for each graph: clique $[1, 2, 3]$ can be destroyed in two distinct ways by rearranging its vertices in the sequences $[2, 1, 3]$ or $[2, 3, 1]$; clique $[4, 5, 6]$ can be destroyed in two distinct ways by rearranging its vertices in the sequences $[5, 4, 6]$ or $[5, 6, 4]$; finally the cliques can be reversed with respect to each other in any cycle including both $[3, 2, 1]$ and $[4, 5, 6]$; this can be done in 20 different ways: $[3, 2, 1, 4, 5, 6]$, $[3, 2, 4, 1, 5, 6]$, $[3, 2, 4, 5, 1, 6]$, and so on. Therefore 24 configurations must be examined on each graph.

In general, for each graph and for each clique pair of size $k = s + 1$, there are $2(\frac{k!}{2} - 1)$ destructive repairs and $\frac{(2k)!}{(k!)^2}$ non-destructive repairs to be evaluated. The cost of each destructive repair is the minimum cost to visit k vertices according to a prescribed sequence, while the cost of a non-destructive repair is the minimum cost to visit $2k$ vertices according to a prescribed sequence.

These costs can be computed as the costs of shortest paths with forbidden vertices. For instance the repair cost corresponding to sequence $[2, 1, 3]$ is the sum the costs of the following shortest paths: between 0 (the depot) and 2 without

visiting any vertex in $\{1, 3\}$; between 2 and 1 without visiting any vertex in $\{0, 3\}$; between 1 and 3 without visiting any vertex in $\{0, 2\}$; between 3 and 0 without visiting any vertex in $\{1, 2\}$. Hence each destructive repair requires $k + 1$ shortest path computations and each non-destructive repair requires $2k + 1$ shortest path computations. However, the number of shortest paths to be considered is not given by the product between the number of permutations to examine and the number of shortest paths to evaluate in each permutation, because all permutations share the same shortest paths, i.e. those connecting two vertices of a subset without visiting the others. So, to evaluate all destructive repairs of a given clique on a given graph $\frac{k(k+1)}{2}$ shortest paths must be computed; to evaluate all non-destructive repairs for a given clique pair on a given graph $(k + 1)^2 + 1$ shortest paths must be computed.

The number of vertex permutations to be examined and shortest paths to be computed is slightly smaller, when the two cliques share a vertex. Therefore, to speed up the bounding procedure, it is profitable to select a clique pair with this property.

In order to possibly iterate the additive bounding procedure, reduced costs can be computed by the shortest path algorithm (Dijkstra algorithm), so that they can be used as edge costs in a subsequent repair step based on different clique pair. However, in our experiments we observed that after repairing the violation represented by the first clique pair, the resulting reduced costs are so close to zero that it is almost never possible to achieve further improvements of the lower bound. Therefore, it is a reasonable option to skip the computation of reduced costs in the shortest path algorithm, to save time.

Heuristic Repair

The bounding procedure can also be accelerated by considering only some clique pairs, pre-selected according to a heuristic criterion estimating how promising they are in terms of the additive lower bound they can provide. Alternatively, one can sort the clique pairs according to some heuristic criterion and stop as soon as a strictly positive (or "large enough") repair cost is found. In our tests, as a heuristic criterion we devised the following: all cliques are listed; the destructive repair cost is computed for each of them; then, the figure of merit associated with each clique pair is the sum of the two corresponding destructive repair costs: the higher such a total cost, the more promising the clique pair; then, all clique pairs are examined until a strictly positive repair cost is found.

3 Experimental Results on Lower Bound Tightness

Our first set of experimental tests was driven by the following questions:

1. Are additive lower bounds tighter than the lower bound given by the two optimal Hamiltonian cycles?
2. How is the trade-off between bound tightness and computing time when the repair step is done in a heuristic way?

Table 1 Lower bounds (best values are bolded) and computing time of the additive bounding procedure. A time limit was fixed at 30 s

Instance	DTSP	ER	HR	ER time (s)	HR time (s)
R00_10_2.dat	648	**652**	**652**	1.361	0.266
R01_10_2.dat	**675**	**675**	**675**	0.088	0.110
R02_10_2.dat	**589**	**589**	**589**	0.099	0.104
R03_10_2.dat	**595**	**595**	**595**	0.133	0.029
R00_12_2.dat	677	**684**	**684**	29.152	0.533
R01_12_2.dat	**708**	707	707	10.397	0.455
R02_12_2.dat	**595**	**595**	**595**	0.247	0.113
R03_12_2.dat	**676**	675	675	0.232	0.145
R00_12_3.dat	677	**682**	**682**	2.229	1.341
R01_12_3.dat	**708**	705	705	1.097	0.361
R02_12_3.dat	**595**	**595**	**595**	0.203	0.062
R03_12_3.dat	**676**	675	675	0.091	0.110
R00_14_2.dat	702	n/a	**707**	30.000	3.183
R01_14_2.dat	716	n/a	**717**	30.000	0.674
R02_14_2.dat	631	n/a	**634**	30.000	2.251
R03_14_2.dat	**732**	n/a	**732**	30.000	1.697

A subset of instances have been extracted from the pool of instances used by Petersen et al. [9]. Table 1 reports the following outcomes:

- DTSP: the value of the double TSP lower bound;
- ER: the lower bound obtained with the exact (exhaustive) repair method;
- HR: the lower bound obtained with the heuristic repair method;
- ER time: the computing time to obtain the bound with exact repair;
- HR time: the computing time to obtain the bound with the heuristic repair.

The last four values in the ER column are marked as *not available* because the time limit of 30 s was reached. They refer to the largest instances of the subset (7×2) for which the exhaustive method could not provide a lower bound. However, the heuristic method succeeded in providing it within the time limit.

The results show that the bound provided by the additive bounding procedure can indeed improve upon the bound given by the double TSP, although the improvement is in average very small. This is mainly due to the gap between the cost of an optimal Hamiltonian cycle and the cost of the Held and Karp lower bound; this difference must be compensated by the repair cost provided by the additive bounds.

This suggests not to run the additive bounding algorithm at every node in the branch-and-bound tree, but only on nodes that are rather deep in the tree, where a small improvement can be enough to close the gap between the upper and the lower bound and the TSP instances are constrained (by branching decisions), so that the Held and Karp lower bound tends to be close to the optimum.

Remarkably, in our tests heuristic repair provided the same lower bounds as exact repair in a fraction of the computing time.

4 Branch-and-Bound

In this section we describe a branch-and-bound algorithm based on the additive bounding method.

4.1 Branching

We devised and tested two branching strategies: the first one is based on the structure of the 1-trees and aims at gradually enforcing Hamiltonian cycle constraints, while the second one is based on the incompatibility graph and aims at gradually enforcing LIFO constraints. Both strategies operate by fixing or forbidding edges. Hence each sub-problem in the branch-and-bound tree is characterized by a set of fixed edges and a set of forbidden edges for each graph.

The effect of fixing or forbidding an edge is obtained by respectively subtracting or adding a "large enough" constant M to the true edge cost.

4.1.1 Branching Policy 1

Branching policy 1 is applicable when at least one of the two 1-trees produced by Held and Karp algorithm is not a cycle. Such a 1-tree certainly contains at least one vertex with degree larger than 2. A branching vertex k is selected among them and the two edges $[k, u]$ and $[k, v]$ belonging to the cycle of the 1-tree are identified. Then three successor sub-problems are generated as follows:

1. $[k, u]$ is forbidden;
2. $[k, u]$ is fixed $[k, v]$ is forbidden;
3. $[k, u]$ and $[k, v]$ are fixed and all the other edges incident to k are forbidden.

Among all the possible choices of k we select the vertex that maximizes the number of non-cycle vertices in the subtree emanating from it.

4.1.2 Branching Policy 2

The final lower additive bound is the cost of a particular permutation among all those that allow to repair an incompatibility represented by a pair of cliques. The repair cost is the sum of some shortest path costs. We consider all the edges in these shortest paths. At least one of them must have strictly positive reduced cost. Given the list of these p positive reduced cost edges, the branching policy generates $p + 1$ sub-problems by fixing one of the edges and forbidding all edges preceding it in the list; the last sub-problem has all the edges forbidden. Branching policy 2 is used only when branching policy 1 is not applicable because both 1-trees are cycles.

4.2 *Lazy Bounding*

Since the most time-consuming part of the branch-and-bound is bounding and in particular computing the repair cost (Step 3), we devised some heuristic criteria to decide whether to run the bounding algorithm or to branch immediately.

In a version of the algorithm, Step 3 is run only when the value $\frac{UB-LB}{UB}$ is below a fixed threshold, where UB and LB indicate the best incumbent upper bound and the lower bound provided by the 1-trees. In case the test does not succeed but the 1-trees are cycles, Step 3 is executed anyway to provide the information needed by branching policy 2. We call this lazy bounding policy "gap repair".

In another version of the algorithm, Step 3 is executed only when both 1-trees computed in Step 2 are cycles. This is motivated by our experiments illustrated in Sect. 3, that suggest that the lower bound increase tends to be almost useless if the 1-tree is not close enough to the optimal Hamiltonian cycle. We call this lazy bounding policy "cycle repair".

5 Experimental Results

In our instances we used as an initial upper bound the best known value (from Alba et al. [1]) and the branch-and-bound tree was explored according to a best-first-search policy.

We compared the three versions of the branch-and-bound algorithm, identified as "always repair", "cycle repair" and "gap repair". The results are shown in Table 2. The threshold for "gap repair" was empirically set to 10%.

"Cycle repair" is the variant that performs best, because the condition for running Step 3 is more restrictive and therefore the additive bounding procedure is completely executed only when it is really able to beat the double cycle lower bound. The difference in computing time in favor of "cycle repair" increases as the size of the instance increases.

Instances larger than those reported could not be solved to proven optimality within 1 h. Therefore the results are not reported here.

Table 2 Processing time of the three variants of the branch-and-bound algorithm

Instance	Always repair	Gap repair	Cycle repair
R00_10_2.dat	35.995	36.224	33.501
R01_10_2.dat	30.220	17.059	29.093
R00_12_3.dat	47.031	47.981	46.290
R01_12_3.dat	91.229	71.845	49.648
R00_12_2.dat	334.122	337.566	297.780
R01_12_2.dat	295.556	304.860	246.236

6 Conclusions

The additive bounding technique can be exploited to compute a lower bound tighter than the one provided by the two optimal cycles for the DTSPMS, owing to its ability of adding the cost to repair at least one of the LIFO constraints violations.

The additive lower bounding algorithm we have devised suffers for the combinatorial explosion in the number of permutations that must be evaluated to repair a violation. It is effective not to pay this effort, especially for non-destructive repair costs for all violations but only for one of them, heuristically selected. The same idea can be further developed in order to limit also the computation of destructive repair costs.

Lazy bounding is effective at substantially reducing the total computing time, especially in its version "cycle repair".

However, to solve larger instances of the DTSPMS it is certainly needed to enrich the approach presented here with further algorithmic ideas to suitably select the clique pairs in the lower bounding procedure and to keep the combinatorial explosion of the number of permutations to consider under control.

After having validated the usefulness of additive bounding for the uncapacitated DTSPMS, another important step that needs to be addressed is the inclusion of capacity constraints.

References

1. Alba Martínez, M.A., Cordeau, J.F., Dell'Amico, M., Iori, M.: A branch-and-cut algorithm for the double traveling salesman problem with multiple stacks. INFORMS J. Comput. **25**, 41–55 (2013)
2. Carrabs, F., Cerulli, R., Speranza, M.G.: A branch-and-bound algorithm for the double travelling salesman problem with two stacks. Networks **61**, 58–75 (2013)
3. Casazza, M., Ceselli, A., Nunkesser, M.: Efficient algorithms for the double traveling salesman problem with multiple stacks. Comput. Oper. Res. **39**, 1044–1053 (2012)
4. Chagas, J.B.C., Toffolo, T.A.M., Souza, M.J.F., Iori, M.: The double traveling salesman problem with partial last-in-first-out loading constraints. Int. Trans. Oper. Res. (2020, to appear). https://doi.org/10.1111/itor.12876
5. Cook, W.: Concorde TSP solver website (2015). http://www.math.uwaterloo.ca/tsp/concorde/index.html
6. Felipe, Á., Ortuño, M.T., Tirado, G.: The double traveling salesman problem with multiple stacks: A variable neighborhood search approach. Comput. Oper. Res. **36**, 2983–2993 (2009)
7. Fischetti, M., Toth, P.: An additive bounding procedure for combinatorial optimization problems. Oper. Res. **37**, 319–328 (1989)
8. Held, M., Karp, R.M.: The traveling-salesman problem and minimum spanning trees. Oper. Res. **18**, 1138–1162 (1970)
9. Petersen, H.L., Madsen, O.B.G.: The double travelling salesman problem with multiple stacks: formulation and heuristic solution approaches. Eur. J. Oper. Res. **198**, 139–147 (2009)
10. Urrutia, S., Milanés, A., Løkketangen, A.: A dynamic programming based local search approach for the double traveling salesman problem with multiple stacks. Int. Trans. Oper. Res. **22**, 61–75 (2015)

Crowd-Shipping and Occasional Depots in the Last Mile Delivery

Luigi Di Puglia Pugliese, Francesca Guerriero, Giusy Macrina, and Edoardo Scalzo

Abstract Crowd-shipping is a new delivery paradigm that is gaining success in the last-mile and same-day delivery process. In crowd-shipping the deliveries are carried out by both regular company vehicles and some crowd-drivers, named occasional drivers (ODs). ODs are ordinary people available to make deliveries, for a small compensation. We consider a setting in which a company not only has ODs available to make deliveries, but they may also use the services of intermediate pickup and delivery points, named occasional depots. In order to optimize the use of these depots, we consider two distinct groups of ODs with different operative ranges. Occasional depots are activated only if it is necessary or convenient; hence, their activation implies an "activation cost". These depots should increase the flexibility of the system and they lead to a more efficient managing of the uncertain availability of ODs. In this work we present a mixed integer linear programming model to represent this framework. We carry out computational experiments to validate it on small size instances.

Keywords Crowd-shipping · Vehicle routing problem · Occasional drivers · Occasional depots · Last mile delivery

1 Introduction

Among the online activities, online shopping is one of the most popular ones worldwide. Always more often people buy online any kind of goods, and have high expectations in delivery service (i.e., high speed, convenient time and place,

L. Di Puglia Pugliese
Istituto di Calcolo e Reti ad Alte Prestazioni, Consiglio Nazionale delle Ricerche, Rende, Italy
e-mail: luigi.dipugliapugliese@icar.cnr.it

F. Guerriero · G. Macrina · E. Scalzo (✉)
Department of Mechanical, Energy and Management Engineering, University of Calabria, Rende, Italy
e-mail: francesca.guerriero@unical.it; giusy.macrina@unical.it; edoardo.scalzo@unical.it

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_18

low cost). Satisfying customers expectations become a key factor, because consumers can easily shop from alternatives retailers. For this reason, finding good strategies for managing same-day/last-mile delivery process (i.e., the transportation of products from a transportation hub to the final delivery destination) is one of the most important activity for the companies. In this context, the larger e-retailers have started to explore unconventional last-mile delivery system to pursue effective and efficient goals. Crowd-shipping is one among the innovative solutions proposed by companies. The main concept of crowd-shipping is to apply the basics of sharing economy to the delivery process, by delegating some deliveries to ordinary people en route to their destination, named occasional drivers (ODs). ODs share the empty space of their own vehicles (car, van, bike) for bringing goods to other people, making some deviations from their ordinary route, for a small compensation. Customers and ODs can register on a dedicated crowd-shipping online platform. By logging on this platform, customers can send the request, check the tracking and confirm the receiving, and ODs can accept the convenient requests, receive all the information to perform the deliveries and finally gain the compensation.

In this work, we propose a crow-shipping delivery system with occasional depots. A fleet of traditional carriers and two types of ODs, with different characteristics and compensation, may perform multiple deliveries in an urban area and/or supply the occasional depots. The main novelty of the proposed framework is the presence of the occasional depots, which makes the deliveries more flexible, then, more attractive for the occasional drivers. In addition, they allow the company to exploit buildings that are already present in the urban area (e.g., shops, kiosks, post offices) without investing in expensive construction/rent of big infrastructures; hence, to assign more deliveries to the occasional drivers, reducing the overall costs. The rest of the paper is structured as follows, in Sect. 2 we describe the state of the art, in Sect. 3 we provide a mathematical programming model for the proposed problem, Sect. 4 depicts the computational results and Sect. 5 summarizes the conclusions and the future directions.

2 State of the Art

ODs are considered for the first time by Archetti et al. [1]. In this work, the authors studied a vehicle routing problem (VRP) variant based on Walmart vision: the fleet is composed not only of company drivers but also of ODs. They named this problem VRP with ODs (VRPOD). In the scenario of this paper the ODs are in-store customers willing to make a delivery for another customer on their way home. Several authors extended the work of Archetti et al. [1]; Macrina et al. [5] introduced time windows constraints, multiple deliveries for ODs and considered also a system with a split delivery policy. Macrina et al. [8] proposed a variable neighborhood search to solve the VRPOD variant with multiple deliveries and time

windows studied by Macrina et al. [5]. Macrina and Guerriero [6] and Macrina et al. [9] introduced a green variant of the VRPOD, while Dahle et al. [3] studied a VRPOD variant with pickup and delivery. Macrina et al. [7] introduced the VRPOD with transshipment nodes. Transshipment nodes are intermediate depots, nearer to the urban area than the main depot. Hence, ODs can pickup goods from the main depot as well as from the transshipment nodes, supplied by the traditional drivers. The authors formulated the problem as a particular instance of two-echelon VRP. Sampaio et al. [10] analyzed the potential benefits of using transfers to support courier deliveries, in particular they highlighted the benefits of using this system when pickup and delivery locations are far apart and the time windows of ODs are tight. All the aforementioned works consider static variants of the VRPOD, i.e., customer demands and ODs availability are known in advance. Recently, some authors considered stochastic/on-line variants of the problem (see, e.g., Dayarian and Savelsbergh [4] and Archetti et al. [2]).

In this paper, we propose a new variant of the VRPOD with occasional depots (ODEs) and two groups of ODs with different functions and compensations. The first group, named classical occasional drivers (CODs), is composed of ODs who are willing to travel for a long range of action. The second group, named neighborhood occasional drivers (NODs), is composed of ODs who are willing to make multiple deliveries in a limited urban area. Hence, one of the main differences between CODs and NODs is the "range of action"; i.e. the number of deviations they are willing to perform. Since NODs have a short range of action and operate in the center of the urban area, they could make deliveries also on foot, by electric vehicles or by bikes. The ODEs can be served by either company drivers or by CODs, and each customer can be served by either company driver or by a NOD. Compensation policy is based on the deviation of the CODs from their route and the routing cost of the NODs. The ODEs could be shops, existing lockers or storage room whose owners don't make the most of their storage capacity. Furthermore, these depots are activated only if convenient, and their activation cost is very cheaper compared to the real transshipment depots. Customers can be visited once, by either an NOD or a company driver. We also suppose that each vehicle can perform a single trip for visiting either customers or occasional depots. The main goal is to serve all customers, overall minimizing costs. The presence of ODEs has four important effects: a higher number of ODs is attracted; it allows a more efficient management of the uncertainty of NODs; it does not involve a construction and management cost; the possibility of activating intermediate depots scattered in the same urban area is very advantageous for efficiently managing the daily customers demand. In addition, the introduction of intermediate nodes, as shown in Macrina et al. [7] and in Sampaio et al. [10], reduces the deviations required to ODs for making the delivery, attracting more ODs. Moreover, the combination of intermediate depots with two groups of ODs adequately rewarded allows to select the drivers closest to the parcel delivery site.

3 Problem Description

Let C be the set of customers, let K be the set of company drivers and let s_0 denote their origin and destination node. It is assumed that each driver may perform a single trip. Let O_1 be the set of available classical occasional drivers that can supply one or more occasional depots from other occasional depots or from s_0. Let O_2 be the set of available neighborhood occasional drivers that can pick up parcels from occasional depots or from s_0 and deliver to customers. Let T be the set of occasional depots that can be served by either the company drivers or the classical occasional drivers. We define the node set as $N := s_0 \cup C \cup T$ and $D := K \cup O_1 \cup O_2$.

Each node pair (i, j) has a positive cost c_{ij} and a travel time t_{ij}. Note that both c_{ij} and t_{ij} satisfy the triangle inequality. Each customer has a single request. Each $d \in D$ has a maximum transport capacity W^d, a maximum number of detours Δ_d (i.e., number of deviations) and an origin node s_d (where his/her trip start) and a destination node t_d. Each $i \in C \cup O_1 \cup O_2 \cup T$ has a time windows $[e_i, l_i]$.

Let x_{ij}^d be a binary variable equal to 1 if and only if a driver d traverses the arc (i, j), and let y_{ij}^{dc} be a binary variable equal to 1 if and only if driver d carries package of customer c from i to j. Moreover, σ_j is a binary variable equal to 1 if and only the occasional depot j is used. Let α_i^d be the arrival time of driver d at node i and let β_j^c be the available time of package of customer c at the occasional depot j. Table 1 reports a summary of the notations.

Table 1 Sets, parameters and decision variables of the proposed model

Sets		Parameters	
s_0	Central depot	ρ_d	Compensation factor
C	Set of customers	τ	Activation cost for ODEs
K	Set of company drivers	M	Large positive number
O_1	Set of CODs	W^d	Maximum transport capacity for driver $d \in D$
O_2	Set of NODs	$[e_i, l_i]$	Time windows of node i
T	Set of ODEs	Δ_d	Maximum number of detours for $d \in O_1 \cup O_2$
N	$N := s_0 \cup C \cup T$ set of all nodes	s_d	Origin node for driver $d \in D$
D	$D := K \cup O_1 \cup O_2$ set of all drivers	t_d	Destination node for driver $d \in D$
		c_{ij}	Travel cost from node i to node j
		t_{ij}	Travel time from node i to node j
Variables			
x_{ij}^d	Binary variable indicating if arc (i, j) is traversed by a driver $c \in D$		
y_{ij}^{dc}	Binary variable indicating if driver $d \in D$ carries package of customer c from i to j		
σ_j	Binary variable indicating if ODE $j \in T$ is activated		
α_i^d	Decision variable specifying the arrival time of the driver $d \in D$ at node i		
β_j^c	Decision variable specifying the time in which the package of customer c is available at ODE $j \in T$		

The problem that aims at minimizing the total cost can be formulated as follows:

$$\min \sum_{d\in D} \rho_d \sum_{i,j\in N} c_{ij} x_{ij}^d - \sum_{d\in O_1} \rho_d \sum_{j\in s_0\cup T} c_{s_d t_d} x_{s_d j}^d + \tau \sum_{j\in T} \sigma_j \tag{1}$$

$s.t.$

$$\sum_{i\in N\cup\{s_d\}} x_{ij}^d \le 1 \qquad \forall d \in D,\, j \in T \cup \{s_0, t_d\} \tag{2}$$

$$x_{ij}^d = 0 \qquad \forall d \in O_1,\, i \in N \cup \{s_d\},\, j \in C \tag{3}$$

$$\sum_{j\in N} x_{s_d j}^d \le 1 \qquad \forall d \in D \tag{4}$$

$$\sum_{i\in N\cup\{s_d\}} x_{ij}^d - \sum_{i\in N\cup\{t_d\}} x_{ji}^d = 0 \qquad \forall d \in D,\, j \in N \tag{5}$$

$$\sum_{i\in N\cup\{s_d\}} \sum_{j\in N} x_{ij}^d \le \Delta_d \qquad \forall d \in O_1 \cup O_2 \tag{6}$$

$$\sum_{c\in C} y_{ij}^{dc} \le W^d \qquad \forall d \in D, i,\, j \in N \tag{7}$$

$$\sum_{d\in D} \sum_{j\in N\setminus\{s_0\}} y_{s_0 j}^{dc} = 1 \qquad \forall c \in C \tag{8}$$

$$\sum_{d\in D} \sum_{i\in N\setminus\{s_0\}} y_{i s_0}^{dc} = 0 \qquad \forall c \in C \tag{9}$$

$$\sum_{i\in N\cup\{s_d\}} \sum_{c\in C} y_{ij}^{dc} = 0 \qquad \forall d \in O_2,\, j \in T \tag{10}$$

$$y_{ij}^{dc} \le x_{ij}^d \qquad \forall d \in D, c \in C, i \in N \cup \{s_d\},\, j \in N \cup \{t_d\} \tag{11}$$

$$\sum_{d\in D} \sum_{i\in N} y_{ij}^{dc} - \sum_{d\in D} \sum_{i\in N} y_{ji}^{dc} = 0 \qquad \forall c \in C,\, j \in N \setminus \{c, s_0\} \tag{12}$$

$$\sum_{i\in N} y_{ij}^{dc} - \sum_{i\in N} y_{ji}^{dc} = 0 \qquad \forall c \in C,\, j \in C \setminus \{c\}, d \in D \tag{13}$$

$$M\sigma_j \ge \sum_{d\in D} \sum_{c\in C} \sum_{i\in N\setminus\{s_0\}} y_{ji}^{dc} \qquad \forall j \in T \tag{14}$$

$$\alpha_i^d + t_{ij} - M(1 - x_{ij}^d) \le \alpha_j^d \qquad \forall d \in D, i \in N,\, j \in N \cup \{t_d\} \tag{15}$$

$$\alpha_i^d \ge e_d + t_{s_d i} - M\Big(1 - \sum_{j\in N} x_{ji}^d\Big) \qquad \forall d \in D, i \in N \tag{16}$$

$$\alpha^d_{t_d} \leq l_d \qquad \forall d \in D \tag{17}$$

$$e_i \leq \alpha^d_i \leq l_i \qquad \forall d \in D, i \in N \cup \{t_d\} \tag{18}$$

$$\alpha^d_j - M\big(1 - \sum_{i \in N} y^{dc}_{ij}\big) \leq \beta^c_j \qquad \forall d \in D, j \in T, c \in C \tag{19}$$

$$\beta^c_j \leq \alpha^d_j + M\big(1 - \sum_{i \in N \setminus \{s_0\}} y^{dc}_{ji}\big) \qquad \forall d \in D, j \in T, c \in C \tag{20}$$

$$x^d_{ij} \in \{0, 1\} \qquad \forall d \in D, i, j \in N \cup \{s_d, t_d\} \tag{21}$$

$$y^{dc}_{ij} \in \{0, 1\} \qquad \forall d \in D, c \in C, i, j \in N \tag{22}$$

$$\sigma_j \in \{0, 1\} \qquad \forall j \in T \tag{23}$$

$$\alpha^d_i \geq 0 \qquad \forall d \in D, i \in N \cup \{t_d\} \tag{24}$$

$$\beta^c_j \geq 0 \qquad \forall c \in C, j \in T \tag{25}$$

The objective function minimizes the total cost, which consists of the routing cost of the company drivers, compensation cost of the occasional drivers, and the activation cost of the occasional depots. Constraints (2)–(6) manage the flows of the drivers. In particular, constraints (2) ensure that all the nodes in N are visited at most once from driver d. Constraints (3) ensure that drivers in O_1 does not serve customers. Constraints (4) ensure that each occasional driver leaves his/her origin node at most once, whereas constraints (5) ensure the flow conservation. Finally, constraints (6) impose a maximum limit of deviations for occasional drivers. Equations (7) are the capacity constraints. Constraints (8)–(13) manage the flows of packages. In particular, constraints (8) ensure that each package come out of central depot s_0, instead constraints (9) ensure that no package may return to the central depot. Constraints (10) ensure that the driver $d \in O_2$ cannot supply an occasional depot. Constraints (11) and (12) ensure that each package is delivered and delivered to corresponding customer, respectively. Constraints (13) ensure that no customer can be used as an occasional depot. Constraints (14) identify the occasional depots used. Constraint (15)–(18) manage the time windows of all the nodes. In particular, constraints (15) compute the arrival time at node j. Constraints (16)–(18) ensure that each customer and each activated occasional depot is visited within its time window. Constraints (19) compute the time at which the package of customer c is available at the occasional depot j; and constraints (20) ensure that the arrival time of the driver d at j is greater than the available time of the package of customer c. Constraints (21)–(25) define the variables domain.

4 Computational Study

In this section, we present the results of our computational tests. The mathematical model was coded in Java and solved to optimality using the commercial solver Cplex 12.10. The experiments were conducted using an 2.6 GHz Intel Core i7-3615Q processor and 8 GB 1600 MHz DDR3 of RAM.

4.1 Generation of Instances

Starting from the VRPODTW instances of Macrina et al. [8], which are based on the classical VRPTW Solomon instances, we generated small VRPODODTW instances. In particular, we considered the sub-class of 10 customers. These instances are composed of 3 company drivers, 10 customers and 3 ODs. We maintained unchanged the features of company drivers and customers. We transformed the ODs into CODs, by considering their coordinates as origins and randomly choosing their destinations, as well as NODs, ODEs and additional CODs are randomly generated. In particular, due to the main characteristics of our framework, we divided into several zones a rectangle that contains all the nodes for each VRPODTW instance. Then, the additional CODs are generated considering the defined zones, the other nodes are derived from the ODs of the original instance.

Let x_m, x_M, y_m, y_M be the minimum and maximum of the abscissas and ordinates of the coordinates of all nodes of the original instance, respectively. We divided the rectangle $R := \{(x, y) \in \mathbb{R}^2 \mid x \in [x_m, x_M],\ y \in [y_m, y_M]\}$ into q equal parts, representing the q neighborhoods. In particular, let s and t be natural numbers such that $s \cdot t = q$ and $s - t$ is minimum with $s \geq t$, then we define for each $i \in \{1, \ldots, q\}$ the neighborhood $Q_i := \{(x, y) \in R \mid x \in [a_x, b_x], y \in [a_y, b_y]\}$, where:

- $a_x := s_i x_M + (1 - s_i)x_m$, where $s_i := \frac{i-1}{s} - \lfloor \frac{i-1}{s} \rfloor$;
- $b_x := (s_i + \frac{1}{s})x_M + (\frac{s+1}{s} - s_i)x_m$;
- $a_y := t_i y_M + (1 - t_i)y_m$, where $t_i := \frac{1}{t} \lfloor \frac{i-1}{s} \rfloor$;
- $b_y := (t_i + \frac{1}{t})y_M + (\frac{t-1}{t} - t_i)y_m$.

We generated 108 VRPODODTW instances composed of q neighborhoods, where $q \in \{2, 4, 6\}$. Moreover, the number of both NODs and ODEs is chosen as a function of q, i.e., we generated instances with either q or $2q$ of each of them. Their coordinates are randomly chosen considering a minimum distance r from the central depot. In particular, $r \in \{0, \bar{r}\}$, where $\bar{r} = \frac{1}{6} \min(x_M - x_m, y_M - y_m)$. In addition, when the number of both NODs and CODs is exactly q, then each of them is placed in each neighborhood, whereas when the number of both NODs and CODs is $2q$, then we place two NODs and two CODs in each neighborhood. The number of CODs is chosen in the set $\{3, 6\}$.

Table 2 Parameter setting

	CDs	$CODs$	$NODs$	$ODEs$
ρ_c	1	0.7	0.6	–
τ	–	–	–	5
W^c	$\|C\|$	10	5	–
Δ_c	$\|C\|$	3	5	–

For the generated CODs and ODEs, we set the time window to $[0, T_{max}]$, where T_{max} is a parameter specified in the original instance. Instead, the time window $[e_d, l_d]$ for a NOD d is generated as follows (see [2]). The first endpoint e_d is randomly chosen in the range $[0, \max(l_i - e_i)/2]$, where $[e_i, l_i]$ are the time windows of the nodes i of the original instance; the second one (i.e., l_d) is randomly generated in the range $[(\tilde{t} + e_d + t_{ds_0}), T_{max}]$, where t_{ds_0} is the time spent by d to go from its origin to the central depot s_0. Whereas, $\tilde{t}$ is a random value in the interval $[\min_{i \in Q_d} t_{s_0 i}, \max_{i \in Q_d} t_{s_0 i}]$, where Q_d is the belonging neighborhood of the origin node of d. Table 2 summarizes the parameter setting. For setting these parameters, we previously carried out a sensitivity analysis considering several combinations of driver compensation and activation cost of occasional depot. We observed that fixing COD compensation and ODE activation cost, and increasing the NOD compensation, more customers are served by company drivers and this implies a significant increase in total cost. Instead, the increase in COD compensation indicates a decrease in the number of occasional drivers and depots used. Thus, it is not convenient to activate the occasional depots when they are supplied by the company drivers only. We set the parameters accordingly.

We grouped the generated instances into four sets, named, A_1, A_2, B_1 and B_2. The sets A_1 and A_2 contain the instances with $q \in \{2, 4, 6\}$ and $r = 0$. The instances belonging to A_1 have q NODs and ODEs, whereas A_2 contains instances with $2q$ of each of them. Both A_1 and A_2 contains 3 CODs. The sets B_1 and B_2 contain the instances with $q = 6$ and $r \in \{0, \bar{r}\}$. The instances belonging to B_1 are characterized by $r = \bar{r}$, whereas B_2 by $r = 0$. In both B_1 and B_2 the number of NODs and ODEs is set to q, and the number of CODs is both 3 and 6. Table 3 summarizes the characteristics of the instances belonging to each group, in terms of number of available occasional drivers, company drivers (CDs), depots and parameter r.

Table 3 Drivers and ODEs available

	r	q	CDs	$CODs$	$NODs$	$ODEs$
A_1	0	2/4/6	3	3	q	q
A_2	0	2/4/6	3	3	$2q$	$2q$
B_1	$\bar{r}$	6	3	3/6	q	q
B_2	0	6	3	3/6	q	q

4.2 Numerical Results

A summary of the results for the instances belonging A_1 and A_2 is shown in Table 4, at varying the parameter q. In particular, 12 instances are considered for each value of q. This table shows the average values related to the number of company drivers and occasional drivers used ($\#CDs$, $\#CODs$, $\#NODs$), the number of occasional depots activated ($\#ODEs$), the run time (in minutes), the total cost, i.e. the value of the objective function ($Cost$), the routing cost of company drivers ($Cost_{CD}$), the detour cost of classical occasional drivers ($Cost_{COD}$), the routing cost of neighborhood occasional drivers ($Cost_{NOD}$) and the activation cost for occasional depots ($Cost_{ODE}$).

As expected, from the analysis of the first two sets of instances, we observe that the higher the number of neighborhoods, the higher the number of customers served by the occasional drivers. As a consequence, the total cost decreases. In particular, when the number of neighborhoods increases, we observe a decrease in the number of company drivers used, a moderate increase in the number of activated ODEs and routed CODs, and a considerable increase of NODs. This can be justified by considering that, the higher the number of the neighborhoods, the higher the number of available ODEs. This aspect allows CODs to perform deliveries with a low deviation from their route.

An ODE is activated and used only if it is necessary and convenient, i.e., there exists at last one NOD that requires to pickup from it. The number of NODs used increases because the number of available and convenient ODEs increases, hence, the cost related to compensation of these drivers decreases.

Comparing the results on A_1 and A_2, on average, we may observe a more evident cost saving for the set A_2. In fact, when increasing the number of neighborhoods, the number of CODs and NODs used increases of about 81%, while focusing on A_2 this number increases more than double. Table 4 shows that, when the number of neighborhoods is increased, almost all costs, all drivers and occasional depots used for the instances belonging to A_2 are subject to a greater variation than that observed for those belonging to A_1, whereas $Cost_{NOD}$ in A_2 is subject to a smaller variation than that in A_1. This is justified by the decrease in the cost for each NOD. In fact, the cost for each of them, on average, has a decrease of 32% and 41% for $q = 6$ with respect to $q = 2$, for the instances belonging to A_1 and A_2, respectively.

In addition, we observe an increase of 38% in A_1 and 15% in A_2 on average in the cost for each COD for $q = 6$ with respect to $q = 2$. It follows that for $q = 6$, the CODs take a longer deviation than that taken for $q = 2$. Whereas, we may observe a decrease of the routing cost for the NODs. Overall, we have a reduction of the total cost for increasing q, since the reduction of $Cost_{NOD}$ suffice the increasing in $Cost_{COD}$.

The average numerical results related to the instances belonging to B_1 and B_2, are reported in Table 5. The column u shows the number of available CODs. For each value of u we considered 12 instances. The superscript near the number in the

Table 4 Average results for the instances belonging to A_1 and A_2 at varying the number of neighborhoods q

Set	q	$\#CDs$	$\#CODs$	$\#NODs$	$\#ODEs$	$Time$	$Cost$	$Cost_{CD}$	$Cost_{COD}$	$Cost_{NOD}$	$Cost_{ODE}$
A_1	2	1.17	0.33	1.75	0.50	4.95	204.95	90.15	2.35	109.95	2.50
	4	0.42	0.42	2.75	0.50	8.34	175.99	35.33	3.64	134.90	2.50
	6	0.25	0.67	3.33	0.75	21.96	165.56	13.88	6.58	140.55	3.75
A_2	2	1.25	0.42	1.75	0.67	33.07	203.59	90.40	3.02	106.83	3.33
	4	0.25	0.75	3.00	0.83	31.04	171.97	22.92	6.27	139.04	4.17
	6	0.00	1.25	3.58	1.50	59.20	149.38	0.00	10.35	131.53	7.50

Table 5 Average results for the instances belonging to B_1 and B_2 at varying the number of available CODs (u)

Set	u	$\#CDs$	$\#CODs$	$\#NODs$	$\#ODEs$	$Time$	$Cost$	$Cost_{CD}$	$Cost_{COD}$	$Cost_{NOD}$	$Cost_{ODE}$
B_1	3	0.33	1.08	3.33	1.17^{2}	9.94	170.15	18.37	6.82	138.29	5.83
	6	0.17	1.92	3.58	2.00^{7}	12.72	162.90	12.79	10.55	129.55	10.00
B_2	3	0.25	0.67	3.33	0.75^{0}	21.96	165.56	13.88	6.58	140.55	3.75
	6	0.25	1.17	3.50	1.25^{5}	10.47	161.84	13.62	8.14	133.82	6.25

column $\#ODEs$ represents the total number of ODEs supplied from other ODEs, in all the considered instances.

The results show, as expected, that increasing the number of CODs available implies both a decrease in the total cost and an increase in the ODEs activated. This behavior can be justified by considering two aspects. First of all, the presence of a high number of CODs increases the possibility of finding convenient ODEs to supply from the central depot. Secondly, the number of ODEs supplied from other ODEs increases as indicated by the superscript numbers in the $\#ODEs$ column. The latter is the most interesting behavior. Indeed, it is possible to find configurations in which an ODE is served even if it is not convenient to activate it individually, because it is cheaper than the NODs delivery to customers directly. The increase in cost to supply a first ODE is compensated by the activation of a second one, which is served starting from the first. In this case, the former is often used by both groups of occasional drivers.

An interesting aspect that we may observe is that in B_1 a higher number of ODEs are used than that used for the instances belonging to B_2. On the one hand, comparing the results obtained for the instances belonging to B_2 with those of B_1, we observe an increase of CODs who have to supply the activated depots. Therefore, there is an increase in the total cost for both the routing of the CODs and the activation of the ODEs. On the other hand, the number of NODs remains almost unchanged but the cost for each of them increases, on average. This is justified by the fact that the activated ODEs for B_1 are closer to the NODs than the activated ODEs for the instances belonging to B_2. Comparing the results obtained for the instances belonging to B_1 and B_2, we may conclude that the farther the NODs are from the central depot, the more it is convenient to encourage the activation of ODEs by increasing the number of CODs. Indeed, the savings on the total cost for the instances belonging to B_1 are more evident than those observed for the instances of B_2.

5 Conclusions

In this paper, we introduced a variant of the vehicle routing problem with occasional drivers and time windows (VRPODTW) where occasional depots (ODEs) are considered. These depots are either shops, existing lockers or small stores located in the urban areas, that are activated only if it is necessary. The fleet of occasional drivers (ODs) is composed of two main classes of drivers with different operations and compensations: drivers belonging to the first one (CODs) carry parcels from the central depot to the activated ODEs, whereas the second one (NODs) are dedicated to serve the customers picking up the parcels from either the central depot or the activated ODEs. It is allowed the CODs to transfer parcels among the ODEs.

We propose a mixed-integer programming model for the problem. We analyze the behavior of the considered transportation system by solving the model on a set of small-size instances generated starting from benchmarks for the VRPODTW.

The results showed that the availability of ODEs allows cost savings. In addition, the presence of both groups of occasional drivers induces an efficient and effective organization of the delivery process. In particular, the routing cost for the NODs decreases for increases number of activated ODEs. In addition, the presence of the CODs allows the replenishment of the activated ODEs profitable. The cost paid to activate the ODEs is sufficed by the efficient usage of both CODs and NODs. Hence, we observe an overall reduction of the transportation cost.

For future work we intend to develop heuristics and exact approaches for the proposed model and to extend the testing on large-size instances.

Acknowledgments This work is supported by MIUR funds, project: "Innovative approaches for distribution logistics"—Code DOT1305451—CUP H28D20000020006.

References

1. Archetti, C., Savelsbergh, M.W.P., Speranza, M.G.: The vehicle routing problem with occasional drivers. Eur. J. Oper. Res. **254**, 471–480 (2016)
2. Archetti, C., Guerriero, F., Macrina, G.: The online vehicle routing problem with occasional drivers. Comput. Oper. Res. **127**, 105144 (2021)
3. Dahle, L., Andersson, H., Christiansen, M., Speranza, M.G.: The pickup and delivery problem with time windows and occasional drivers. Comput. Oper. Res. **109**, 122–133 (2019)
4. Dayarian, I., Savelsbergh, M.: Crowdshipping and same-day delivery: employing in-store customers to deliver online orders. Prod. Oper. Manag. **29**(9), 2153–2174 (2020)
5. Macrina, G., Di Puglia Pugliese, L., Guerriero, F., Laganà, D.: The vehicle routing problem with occasional drivers and time windows. In: Sforza, A., Sterle, C. (eds.) Optimization and Decision Science: Methodologies and Applications, volume 217 of Springer Proceedings in Mathematics & Statistics, pp. 577–587. ODS, Sorrento. Springer, Cham (2017)
6. Macrina, G., Guerriero, F.: The green vehicle routing problem with occasional drivers. In: Daniele, P., Scrimali, L. (eds.) New Trends in Emerging Complex Real Life Problems. AIRO Springer Series, vol. 1. Springer, Cham (2018)
7. Macrina, G., Di Puglia Pugliese, L., Guerriero, F.: A variable neighborhood search for the vehicle routing problem with occasional drivers and time windows. In: ICORES 2020 - Proceedings of the 9th International Conference on Operations Research and Enterprise Systems, pp. 270–277 (2020)
8. Macrina, G., Di Puglia Pugliese, L., Guerriero, F., Laporte, G.: Crowd-shipping with time windows and transshipment nodes. Comput. Oper. Res. **113**, 104806 (2020)
9. Macrina, G., Di Puglia Pugliese, L., Guerriero, F.: Crowd-shipping: A new efficient and eco-friendly delivery strategy. Procedia Manuf. **42**, 483–487 (2020)
10. Sampaio, A., Savelsbergh, M., Veelenturf, L.P., Van Woensel, T.: Delivery systems with crowd-sourced drivers: A pickup and delivery problem with transfers. Networks **76**, 232–255 (2020)

Branch and Bound and Dynamic Programming Approaches for the Path Avoiding Forbidden Pairs Problem

Daniele Ferone, Paola Festa, and Matteo Salani

Abstract We propose a branch and bound (B&B) and a dynamic programming algorithm for the Path Avoiding Forbidden Pairs Problem (PAFPP). Given a network and a set of forbidden node pairs, the problem consists in finding the shortest path from a source node s to a target node t, avoiding to traverse both nodes of any of the forbidden pairs. The problem has been shown to be NP-complete. In this work, we describe the problem, its mathematical model and we propose two exact algorithms. We compare their performances against those of a commercial solver solving instances for two different graph topologies: fully random graphs and grid graphs.

Keywords Branch and bound · Dynamic programming · Constrained shortest paths · Forbidden pairs

1 Introduction

In this work, we study the *Path Avoiding Forbidden Pairs Problem* (PAFPP). The problem asks to find the shortest path between two nodes s and t in a given weighted directed graph $G = (V, A)$ or recognize that such path does not exist. The shortest path should not visit both nodes belonging to a set $F \subset (V \times V)$ of node pairs called

D. Ferone (✉)
Department of Mechanical, Energetic and Management Engineering, University of Calabria, Rende, Italy
e-mail: daniele.ferone@unical.it

P. Festa
Department of Mathematics and Applications, University of Napoli FEDERICO II, Naples, Italy
e-mail: paola.festa@unina.it

M. Salani
Dalle Molle Institute for Artificial Intelligence, USI-SUPSI, Lugano, Switzerland
e-mail: matteo.salani@idsia.ch

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_19

forbidden pairs. Paths containing at most one vertex from each pair in F are called F-paths.

The PAFPP has been introduced in [12, 15] to design test cases for automatic software validation, where the nodes of the graph represent segments of code and edges represent the control flow. The goal is to cover the graph with $s - t$ paths corresponding to different test cases, introducing forbidden pairs which identify the mutually exclusive code segments.

A special case of PAFPP arises in bio-informatics, tackling the problem of peptide sequencing via tandem mass spectrometry. Chen et al. [3] model the peptide sequencing as a PAFPP on a directed acyclic graph.

PAFPP emerges in Ferone et al. [6] as a subproblem of the constrained shortest path problem named *Constrained Shortest Path Tour Problem (CSPTP)*. Authors reduce the CSPTP to the PAFPP and solve it with a branch and bound strategy.

Lastly, Ceselli et al. [2] solve a rich Vehicle Routing Problem using Branch and Price. Here the PAFPP structure emerges in the pricing problem modeling compatibility constraints between customers.

Gabow et al. [9] proves the NP-hardness of PAFPP, which is polynomially solvable under *skew symmetry* conditions [16]. Kolman and Pangrác [10] studies the complexity of PAFPP under different assumptions, showing that the problem remains NP-complete even if the graph is planar or presents an halving structure, but it becomes polynomial when the graph has a hierarchical structure. These results are extended in [11], proving that the PAFPP is NP-hard when the set of forbidden pairs has an overlapping structure or is sorted. Finally, Blanco et al. [1] presents a polyhedral study of the PAFPP.

The paper is organized as follows. In Sect. 2 we present the mathematical model of the problem. In Sects. 3 and 4 we present the solution approaches and the computational results, respectively. In Sect. 5 we conclude and discuss some future work.

2 Mathematical Formulation

Let $G = (V, A)$ be a weighted directed graph, where $V = \{1, \ldots, n\}$ is the set of nodes, and $A = \{(i, j) \in V \times V : i, j \in V \wedge i \neq j\}$ is the set of m arcs. Let $C : A \rightarrow \mathbb{R}_0^+$ a function that assigns a non-negative cost c_{ij} to each arc $(i, j) \in A$. For each node $i \in V$, let $FS(i) = \{j \in V : (i, j) \in A\}$ and $BS(i) = \{j \in V : (j, i) \in A\}$ be the forward star and backward star of node i, respectively. Given a source node $s \in V$ and a destination node $t \in V$, the PAFPP can be modeled with the following $0 - 1$ integer program:

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{1a}$$

s.t.

$$\sum_{j \in FS(i)} x_{ij} - \sum_{j \in BS(i)} x_{ji} = \begin{cases} 1, & i = s; \\ -1, & i = t; \\ 0, & \text{otherwise}; \end{cases} \tag{1b}$$

$$\sum_{j \in BS(a)} x_{ja} + \sum_{j \in BS(b)} x_{jb} \leq 1 \qquad \forall\, (a, b) \in F \tag{1c}$$

$$x_{ij} \in \{0, 1\} \qquad \forall\, (i, j) \in A. \tag{1d}$$

The objective function (1a) minimizes the path length. Constraints (1b) model the flow balance at each node. Constraints (1c) guarantee that two nodes belonging to a forbidden pair are never visited simultaneously.

3 Solution Approaches

In this section, we describe two exact approaches to solve the PAFPP. In particular, we present a branch and bound algorithm (B&B) in Sect. 3.1, and a dynamic programming algorithm in Sect. 3.2.

3.1 Branch and Bound Approach

Observing the mathematical model (1a)–(1d), it is evident that relaxing the constraints (1c) we obtain the model of a Shortest Path Problem (SPP), which is polynomially solvable (for example, with the Dijkstra's algorithm [5]).

Therefore, we devise a B&B algorithm using a polynomial algorithm for the SPP to compute a combinatorial bound and performing branching operations when the optimal solution of the relaxation results infeasible for the PAFPP.

Let G^t be the graph associated to a generic iteration t of the B&B, let P^t be the optimal solution of the relaxed problem PAFPP ^t_R on G^t and let UB be the value of an incumbent feasible solution. If the value of P^t is not less than UB, then the graph G^t does not contain any improving solution and P^t can be disregarded. Instead, if P^t does not contain any forbidden pair, then it is also feasible for PAFPP defined on G and it improves the incumbent solution. On the contrary, if P^t contains both nodes of any forbidden pair, two sub-problems are generated.

In particular, let (v, w) one of the forbidden pairs violated by P^t, two graphs G^{t^1} and G^{t^2} are generated and associated to the branching nodes t^1 and t^2, respectively. The graphs G^{t^1} and G^{t^2} are obtained removing from G^t the nodes v and w, respectively. More formally, $A^{t^1} = A^t \setminus \{v\}$ and $A^{t^2} = A^t \setminus \{w\}$.

Obviously, as in classic B&B framework, if the sub-problem of iteration t is not feasible – i.e., it does not exist an $s - t$ path due to the removed nodes—the node is

not further branched. The incumbent best solution found is optimal and returned as final solution.

3.2 Dynamic Programming for PAFPP

Dynamic programming have been extensively and successfully applied to constrained shortest path problems [4, 7]. Therefore, we solve PAFPP to optimality by a bi-directional dynamic programming algorithm [13] implementing the Decremental State Space Relaxation (DSSR) strategy [14].

A state associated with vertex $i \in N$ represents a partial path from the source node s to the node i. Different states can be associated with the same node and they correspond to different partial paths.

The dynamic programming algorithm iteratively extends states until no further extensions are possible. Among all feasible states reaching the destination node d the one with minimal cost represent the optimal solution to PAFPP.

Each state is encoded in a label, in bi-directional dynamic programming called *forward* and *backward* labels. A forward label associated with node $i \in N$ is a tuple:

$$l_i^f = (i, c_i, S, B), \tag{2}$$

where i is the last node visited in the partial path, c_i is the accumulated cost, S is a binary vector that keep tracks of the visited nodes in the partial path and B is a binary vector with size $|B| = |F|$. In vector B, $b_j \in B = 1$ indicates that one of the nodes belonging to the jth forbidden pair is visited along the partial path. Note that, S does not keep any information about the order in which the vertices are visited. Similarly, a backward label associated with node $i \in N$, corresponding to paths from node i to destination node d, is a tuple:

$$l_i^b = (i, c_i, S, B), \tag{3}$$

where tuple's elements have the same meaning as those of forward labels.

The dynamic programming algorithm extends all feasible forward and backward labels to generate new forward and backward labels. The extension of a forward label corresponds to appending an additional arc (i, j) to a path from s to i, obtaining a path from s to j, while the extension of a backward label corresponds to appending an additional arc (j, i) to a path from i to d, obtaining a path from j to d.

The binary vector B is used to avoid visiting pair of nodes belonging to a forbidden pair while vector S is used to avoid cycles. Anyway, as the cost matrix is non-negative and triangular inequality holds, all partial paths with cycles are suboptimal and can be safely ignored. When a label $l_i = (i, c_i, S, B)$ is extended to a vertex j, a new label $l_j = (j, c_j, S', B')$ is generated. The update rules of the

vectors S and B are as follows:

$$S'_k = \begin{cases} S_k + 1, & k = j; \\ S_k, & k \neq j; \end{cases} \tag{4}$$

$$B'_k = \begin{cases} B_k + 1, & (a,b)_k \in F,\ a_k = j \vee b_k = j; \\ B_k, & \text{otherwise.} \end{cases} \tag{5}$$

A label $l_i = (i, c_i, S, B)$ is feasible if $S_k \leq 1$ and $B_f \leq 1$ for all $k \in N$ and all $f \in F$, respectively.

The effectiveness of dynamic programming depends on the number of generated labels. In order to control the number of labels, dominance tests are performed. Let $l' = (i, c'_i, S', B')$ and $l'' = (i, c''_i, S'', B'')$ be two labels associated with node i. l' dominates l'' and label l'' can be safely discarded only if

$$c'_i \leq c''_i; \tag{6}$$

$$B'_f \leq B''_f, \quad \forall f \in F \tag{7}$$

and at least one inequality is strict. Please note that only vector B participates in the domination criterion.

In bi-directional dynamic programming forward and backward labels are joined to produce complete paths from node s to node d. Let $l_i^f = (i, c_i^f, S^f, B^f)$ a forward label and $l_i^b = (i, c_i^b, S^b, B^b)$. The join is feasible if

$$S_k^f + S_k^b \leq 1, \quad \forall k \in N; \tag{8}$$

$$B_f^f + S_f^b \leq 1, \quad \forall f \in F. \tag{9}$$

The join condition ensures that the final path does not contain cycles nor visit both nodes of a forbidden pairs. Even if the vector S is not included in domination conditions, it is guaranteed that none of the optimal paths is eliminated. Indeed, suppose that a join operation is prohibited because a node k is visited in both forward and backward labels. As the cost matrix is positive and triangular inequality holds, there must be non dominated forward and backward labels where node k is not visited and the join is feasible and more profitable.

We reduce the number of labels by selecting a monotone resource and extend labels for which the resource consumption is less than half of a given threshold T. In PAFPP, the only available monotone resource is the accumulated cost. In order to apply the bi-directional dynamic programming algorithm we compute an upper bound to the optimal cost $\bar{c^*}$ as the threshold T.

Decremental state space relaxation (DSSR) introduced by Righini and Salani [14] aims at reducing the number of states to be explored by dynamic programming. For PAFPP, the basic idea is that not all forbidden pairs are tracked in the vector B

and are therefore not imposed in the domination criterion. If the optimal solution visits both nodes of a forbidden pair, the corresponding pair is added to the vector B and the process is iterated. We remark that at each iteration a lower bound to the optimal solution is computed.

4 Computational Results

We present some preliminary results to compare the performances of DSSR and B&B against those of a commercial solver (IBM CPLEX) directly solving the model (1a)–(1d). A time limit of 10 min has been used for each solution method.

The instances were randomly generated through an adaption of the generator presented in [8] and can be divided in two classes: fully random and grid graphs.

The number m of edges in the random graphs has been selected to be in $\{5 \cdot n, 10 \cdot n, 15 \cdot n\}$, where $n = |V|$. The total number of forbidden pairs for each instance is in $\{25 \cdot n, 30 \cdot n, 35 \cdot n\}$. Since the grid graphs are more sparse, the number of forbidden pairs in grid graphs ranges in $\{\lfloor 6.25 \cdot n \rfloor, \lfloor 12.50 \cdot n \rfloor, \lfloor 18.75 \cdot n \rfloor, \lfloor 25 \cdot n \rfloor, \lfloor 30 \cdot n \rfloor\}$.

Each combination of graph size characterizes a collection of similar instances, denoted as $\{R1, \ldots, R9, G1, \ldots, G3\}$. Each random (grid) collection contains 30 (50) different instances of the same type, 10 for each different number of forbidden pairs. The characteristics of the data-set are summarized in Table 1.

The computational results obtained by CPLEX, B&B, and the dynamic programming approach (DSSR) are reported in Table 2. For each instance type, we report the time spent by the algorithms in solving instances of that type (avg. time), and the number of instances of that type for which a proved optimal (O) solution has been found.

The results highlight that in spite of their size the random graphs are much easier to solve respect to the grid networks. This was an expected result, since random graphs are denser respect to the grid graphs, it is therefore much easier to find an

Table 1 Instance parameters

Fully random graphs			Grid graphs	
Problem	Nodes	Arcs	Problem	size
R1	1500	7500	G1	200 × 200
R2	1500	15,000	G2	200 × 400
R3	1500	22,500	G3	300 × 300
R4	2000	10,000		
R5	2000	20,000		
R6	2000	30,000		
R7	2500	12,500		
R8	2500	25,000		
R9	2500	37,500		

Table 2 Experimental results

	CPLEX		B&B		DSSR	
	O	Avg. time	O	Avg. time	O	Avg. time
G1	27	363.28	18	387.05	10	481.58
G2	28	382.40	30	314.45	14	452.67
G3	23	362.71	27	283.32	15	434.85
Average	*26*	*369.46*	*25*	*328.27*	*13*	*456.37*
R1	30	1.35	30	0.00	30	0.00
R2	30	2.89	30	0.00	30	0.01
R3	30	4.63	30	0.00	30	0.01
R4	30	1.97	30	0.00	30	0.01
R5	30	3.94	30	0.00	30	0.01
R6	30	6.46	30	0.00	30	0.01
R7	30	2.48	30	0.00	30	0.01
R8	30	5.32	30	0.01	30	0.01
R9	30	8.59	30	0.01	30	0.02
Average	*30*	*4.18*	*30*	*0.00*	*30*	*0.01*

alternative path that does not violates any forbidden pair. For random graphs, all the algorithms are very fast (less than 5 s in average). Both B&B and DSSR get the optimal solution for all the instances. It is worthy to note that all the instances were generated to be non-trivial, i.e. the simple shortest path $s - t$ contains at least one forbidden pair.

On the other hand, the grid graphs are more challenging. In this case, CPLEX misses the optimum in 54 out of 150 cases, B&B is not able to find the optimal solution in 48 cases, and DSSR fails in 111 cases. This is caused by the sparsity of the grid graphs that induces a lower number of feasible $s - t$ paths.

Figure 1 illustrates the number of optimal solutions found with respect to the ratio of forbidden pairs over the number of nodes in the network. The B&B and CPLEX approaches show a decreasing trend: for an increasing number of forbidden pairs, the instances become more challenging and, generally, the methods find less optimal solutions. Instead, the DSSR does not seem to be strongly influenced by the number of forbidden pairs as no clear trend is visible.

These are preliminary results, but they give valuable information. When instances are not extremely challenging (Random), both B&B and DSSR perform well presenting an high converge speed to the optimum. Meanwhile, on sparse graphs B&B has similar performances with respect to CPLEX, but solving a higher number of instances. DSSR is the worst approach on these problems. Nevertheless, we believe that DSSR can be strongly improved with the use of an upper bound that permits to prune many feasible labels.

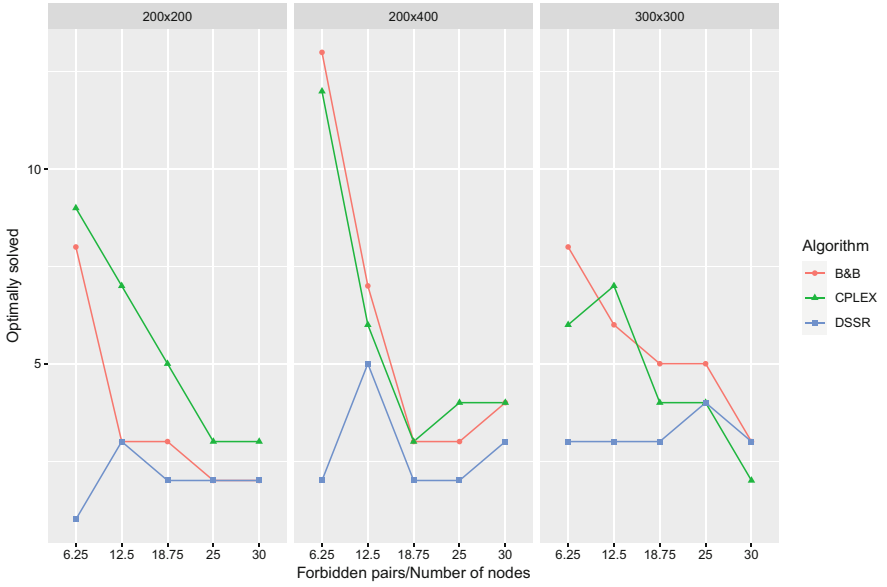


Fig. 1 Optimal solutions found on grid graphs with respect to the relative number of forbidden pairs

5 Conclusions and Future Work

This paper presents the Path Avoiding Forbidden Pairs Problem (PAFPP). We propose two exact algorithms to solve the problem to proven optimality: a branch and bound (B&B) algorithm and a dynamic programming (DSSR) algorithm. Some preliminary results are compare the performance of the methods against those of a commercial solver. The results evidence that on Random instances the two approaches are very performing. On the other hand, on sparse instances B&B seems to be equivalent to CPLEX while DSSR needs to be improved.

As future research perspectives, we are sure to be able to obtain a better upper bound to improve the performance of DSSR and we plan to better investigate the instances' properties that have an impact to the performance of the algorithms.

References

1. Blanco, M., Borndörfer, R., Brückner, M., Hoàng, N.D., Schlechte, T.: On the path avoiding forbidden pairs polytope. Electron. Notes Discrete Math. **50**, 343–348 (2015). https://doi.org/10.1016/j.endm.2015.07.057
2. Ceselli, A., Righini, G., Salani, M.: A column generation algorithm for a vehicle routing problem with economies of scale and additional constraints. Transp. Sci. **43**(1), 56–69 (2009). https://doi.org/10.1287/trsc.1080.0256

3. Chen, T., Kao, M.Y., Tepel, M., Rush, J., Church, G.M.: A dynamic programming approach to de novo peptide sequencing via tandem mass spectrometry. J. Comput. Biol. **8**(3), 325–337 (2001). PMID: 11535179. https://doi.org/10.1089/10665270152530872
4. Di Puglia Pugliese, L., Ferone, D., Festa, P., Guerriero, F.: Shortest path tour with time windows. Eur. J. Oper. Res. **282**(1), 334–344 (2020). https://doi.org/10.1016/j.ejor.2019.08.052
5. Dijkstra, E.: A note on two problems in connexion with graphs. Numer. Math. **1**(1), 269–271 (1959). https://doi.org/10.1007/BF01386390
6. Ferone, D., Festa, P., Guerriero, F., Laganà, D.: The constrained shortest path tour problem. Comput. Oper. Res. **74**, 64–77 (2016). JCR. https://doi.org/10.1016/j.cor.2016.04.002
7. Ferone, D., Festa, P., Fugaro, S., Pastore, T.: A dynamic programming algorithm for solving the k-color shortest path problem. Optim. Lett. (2020). https://doi.org/10.1007/s11590-020-01659-z
8. Festa, P., Pallottino, S.: A pseudo-random networks generator. Tech. rep., Department of Mathematics and Applications ''R. Caccioppoli'', University of Napoli FEDERICO II (2003)
9. Gabow, H.N., Maheshwari, S.N., Osterweil, L.J.: On two problems in the generation of program test paths. IEEE Trans. Softw. Eng. **SE-2**(3), 227–231 (1976). https://doi.org/10.1109/TSE.1976.233819
10. Kolman, P., Pangrác, O.: On the complexity of paths avoiding forbidden pairs. Discrete Appl. Math. **157**(13), 2871–2876 (2009). https://doi.org/10.1016/j.dam.2009.03.018
11. Kováč, J.: Complexity of the path avoiding forbidden pairs problem revisited. Discrete Appl. Math. **161**(10–11), 1506–1512 (2013). https://doi.org/10.1016/j.dam.2012.12.022
12. Krause, K., Goodwin, M., Smith, R.: Optimal software test planning through automated network analysis. TRW Systems Group (1973)
13. Righini, G., Salani, M.: Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. Discrete Optim. **3**(3), 255–273 (2006). https://doi.org/10.1016/J.DISOPT.2006.05.007
14. Righini, G., Salani, M.: New dynamic programming algorithms for the resource constrained elementary shortest path problem. Networks Int. J. **51**(3), 155–170 (2008). https://doi.org/10.1002/net.20212
15. Srimani, P.K., Sinha, B.P.: Impossible pair constrained test path generation in a program. Inf. Sci. **28**(2), 87–103 (1982). https://doi.org/10.1016/0020-0255(82)90019-6
16. Yinnone, H.: On paths avoiding forbidden pairs of vertices in a graph. Discrete Appl. Math. **74**(1), 85–92 (1997). https://doi.org/10.1016/S0166-218X(96)00017-0

Revenue Management Approach for Passenger Transport Service: An Italian Case Study

Francesca Guerriero, Martina Luzzi, and Giusy Macrina

Abstract The main aim of the revenue management (RM) techniques is to sell the right product to the right customer, at the right time and price, trying to anticipate customers' actions, hence, optimizing the sales. RM has been successfully applied in numerous industries, such as airlines, hotels, car rentals, ferry lines, and cruise. In this work, we focus on the passenger transport service. Even though, this industry presents general characteristics to an effective and efficient implementation of a RM process, RM in bus passenger transport has received limited attention. Recently this sector has been deregulated and liberalized, thus bus companies are free to vary their prices, timetable, routes, and services. Hence, the use of RM represents a key factor for companies that must operate in a highly competitive market. We consider the problem of a bus transport company which provides transport service from a given set of origins to a given set of destinations, on a given time horizon. We define a dynamic programming formulation and a linear approximation for the problem under study. The proposed linear approximation, representing the bus seat assignment problem), is tested empirically on the basis of an extensive computational phase with reference to an Italian bus company. The computational experiments reveal that the proposed model could help the bus transport company to control the capacity levels, to improve customer service and bus utilization, by maximizing the revenue.

Keywords Revenue management · Passenger transport · Seat allocation problem

F. Guerriero · M. Luzzi · G. Macrina (✉)
Department of Mechanical, Energy and Management Engineering, University of Calabria, Rende, Italy
e-mail: francesca.guerriero@unical.it; lzzmtn97p41c351k@studenti.unical.it; giusy.macrina@unical.it

R. Cerulli et al. (eds.), *Optimization and Decision Science*, AIRO Springer Series 7,
https://doi.org/10.1007/978-3-030-86841-3_20

1 Introduction

The ever-increasing and continuous movement of people around the world for various purposes (e.g., study, work, pleasure) and with different transport modes (i.e., train, airplane, bus, car) has caused a widely expansion of passenger transport service, both public and private. Since mobility is central to the whole of society, offering very efficient and effective services to customers is a critical issue for all the companies operating in this sector. Disruptive innovations in passenger transport service can be treated in different ways. There are techniques based on business model theories (i.e., Flixbus in the bus sector or Uber in the car one) and others based on revenue management (RM) approaches, both are catalysing the research in this field and are paving the way to new paradigms of business in passenger transport [1].

In this paper we focus on the bus market. The bus demand has significantly increased in recent years, and the revenue of the bus segment is projected to reach 24,034m US dollars in 2025 (www.statista.com). The regularization of the bus transport sector strongly affected the growth of the bus market after 2005. The main intent of the Transport Regulation Authority (ART) has been to regulate the buses transport services periodically offered to people, for connecting more than two regions. Before the legislative decree (n. 285/2005), that replaced the previous law (28 September 1939, no. 1822), a particular line/service was provided exclusively by one company and no other ones could offer the same service within a radius of 30 km; hence, there was an exclusive service concession regime. After the regularization and liberalization, a concession-based market has been introduced; hence, a service could be provided by several companies and it is not longer exclusive. After the liberalization, on the one hand companies have started to propose their transport plans, showing a wide variety of service offers; on the other one, the demand has started to growth, revealing the great potential of this sector. The Italian market has opened up to other world-renowned competitors operating in long-distance bus market, which have introduced new business models and low cost prices. This resulted in a significant increase in the offer and a global lowering of prices, as well as a substantial growth of quality in service. Meantime, the national operators had to face off the increase in the number of competitors from all over Europe. In this context, in order to gain both market share and revenue maximization, the implementation of RM policies plays a crucial role. We address the problem of assigning bus seats to different customers, with booking requests arising dynamically and randomly with time, with the aim of maximizing the expected revenue. A dynamic programming formulation is developed to represent the problem under study. To handle the curse of dimensionality, we also propose and empirically validate a linear programming approximation, that is a bus seat allocation problem.

The rest of the paper is structured as follows: Sect. 2 describes the state-of-art on RM and the seat allocation problem in the case of bus passengers transport service. Section 3 presents a dynamic programming formulation for the seat allocation

problem, whereas a description of the proposed linear programming approximation (i.e., the bus seat assignment problem) is provided in Sect. 4. Section 5 discusses the computational experiments, and finally, Sect. 6 summarizes the conclusions and provides the future directions.

2 Revenue Management and Seat Allocation Problem

In this section, we firstly review several works addressing the RM and its applications; secondly, we analyse the state of the art related to the seat allocation problem.

2.1 Revenue Management

The deregulation of U.S. airline industry was a key event in the development of RM methodology. Before the approval of the Airline Deregulation Act in 1978, airlines operated in a tightly regulated environment; controlled by governments and the U.S. Civil Aviation Board (CAB) [14]. After the deregulation, the airlines were free to vary their prices, timetables, routes and services, without requiring any authorization from the CAB, with a consequently increase in competition.

In this new framework, the formulation of models aimed at optimally selling a fixed and perishable inventory within a given time horizon, has become an important issue. Nowadays, the main challenge is to define integrated RM systems including increasingly sophisticated approaches in order to model, estimate and forecast demand, as well as optimize subsequent management decisions with a high level of automation. In [9] a review of the most recent RM applications is provided. In particular, the authors classified and generalized the areas that have received most attention in the following main categories: opaque products, flexible products, upgrading, overbooking, penalisation and risk aversion. In the sale of opaque products, some characteristics of the products are kept hidden until the sale is completed (for example the seat on the plane or the name of the hotel), allowing to attract more demand and implementing price discrimination to increase revenues (see, [6]). Flexible products refer to the specific situation, in which, for each product, a set of equivalent alternatives is available (see, [4]). Thus, the seller can assign an alternative product to the purchaser close to the service time. This allows to delay the decision of assigning a given element, characterizing the product, until the levels of uncertainty are sufficiently low compared to the levels of future demand; with a consequent optimization of the capacity utilization. Upgrading allows the seller to satisfy the demand for a lower quality product with a higher quality one, selected from a series of hierarchically ordered substitutes. Usually, an upgrade is provided at no additional cost to the customer. Therefore, they differ from the practice of upselling (or paid upgrades) that pushes the customers to buy high quality products at a discounted price. Upgrading is often used in the car rental

market (see, [5]). Overbooking is one of the oldest RM practice, widely used in passenger air transport (see, [11]). It consists in selling more products (i.e., seats in the passenger air transport) than the available capacity. A penalty has to be paid if the customer's demand is not satisfied. Personalization consists in proposing to the customer a targeted offer based on his/her characteristics, collected through particular analysis and data mining techniques (see, [17]). Finally, RM is also used for making decisions that minimize certain risk measures.

2.2 Seat Allocation Problem

The application experience shows that RM is a valuable management method for the enhancement of transport efficiency [15]. Since the quantity of resources (i.e., the seats) is limited, maximizing the revenue by optimizing the use of the capacity-constrained resources is a major challenge for transport companies. Introducing RM techniques in passengers transport sector could be a useful strategy to maximize the revenues, by determining the optimal quantity of seats to be allocated. In the airline industry this problem is also known as the booking limit control problem, and it was widely studied in this application field (see, e.g., [3, 12]). Recently, some contributions related to the use of RM techniques in the rail sector have been published. Some authors assume a deterministic passenger demand (see, e.g., [10, 16]), other analyze a probabilistic demand (see, e.g., [2, 16]). Few studies address the joint rail pricing and seat allocation problems (see, e.g., [7] and [8]). Scarce attention has been devoted to the bus market. Even if the management of seats in airline sector is very different than that of railway and bus, the allocation problems studied with reference to the airline and railway sectors could be adapted to handle the main features of the bus transport market. However, it is crucial identifying the main differences among these three applications. The airline and railway have more classes, characterized by different prices. In the bus transport there is often a unique class of service instead. In addition, in airline each leg is single (i.e., it has a unique boarding and a unique landing), this assumption is not reasonable for railway and bus. In bus and railway applications, there are often more stops along the same line, in other words we must control seat capacity by taking into account multi time stage and capacity changes. Hence, a bus seat allocation problem is a multi leg single fare problem that has the purpose of allocating the demand of seats for a set of legs in order to maximize the overall revenue. The term "single fare" denotes that every tickets for a seat, for a specific leg, on a specific line, has an own price. A recent scientific contribution on seat allocation problem for bus transport is [13], which proposes a methodology for optimizing the operational integration of multiple bus lines, to address the spatial non-uniformity of passenger demand. The authors applied five operational strategies: full-route operation, short turn, limited stop, deadheading, and a mixture of either two or three of the latter three strategies. They tested and compared their strategies on a real-case study,

demonstrating that the effectiveness of combined strategies is higher than that of any single strategy.

3 A Dynamic Programming Formulation for the Seat Allocation Problem

We consider a passenger transport company that offers a transport service from a given set of origins to a given set of destinations. The company sells a set of products to several customers on a given time horizon. In the considered scenario, we define a product as an origin-destination (OD) transport service, performed by a bus. At each time of the planning horizon, the company has to decide how to manage the overall capacity in the most profitable way, taking into account that complete information on the future demand is not available.

Let $I = \{i_1, \ldots, i_n\}$ denote the set of n origins and $J = \{j_1, \ldots, j_m\}$ the set of m destinations. A generic product is denoted as the pair $\{(i, j) : j > i, i \in I, j \in J\}$ and represents the OD transport service from the bus station i to the bus station j. All the products are sorted in increasing order of the origin bus station and stored in the set (IJ), that is $(IJ) = (i_1, j_1), \ldots, (i_1, j_m), (i_2, j_1), \ldots, (i_n, j_m)$. The products offered by the company can be indexed as $p = (1, \ldots, |(IJ)|)$. We assume that the company performs the transport service using a set of buses, each of them characterized by a given seating capacity. The buses spread over K lines, $k = 1, \ldots K$ and C^k represents the capacity of line k that is the seating capacity of the bus, which runs on the line k. Each line k consists of a given number of stops denoted as $S_k + 1$ including the starting and the terminal bus stations and S_k legs between each two bus stations. All the products (i.e., OD transport service) produced by each line k, $k = 1, \ldots, K$ are stored in a sequence according to the incremental order of j and i, that is $(IJ)^k = (1, 2)^k, \ldots, (1, S_k + 1)^k, (2, 3), \ldots, (S_k, S_k + 1)$. After numbering all the products in $(IJ)^k$ from left to right, we can get the product sequence indexed by the serial number $p^k = (1, \ldots, |(IJ)^k|)$.

Let $B^k = b^k_{sp^k}, s = 1, \ldots, S^k, p^k = 1 \ldots, |(IJ)^k|$ denote a binary matrix, each element being equal to 1 if product p^k generated by the line k uses leg s and zero otherwise. Each column of matrix B^k contains all the information related to the legs involved in the OD transport services provided by the line k. An example of matrix B^k for a line with $S^k = 4$ stops and 6 products is reported in Fig. 1.

We assume that different lines can deliver the same OD transport service from i to j, that is alternative products are available. Thus, in order to handle this specific

Fig. 1 Representation of the matrix B, for a line with four stops and six products

Leg	(1,2)	(1,3)	(1,4)	(2,3)	(2,4)	(3,4)
(1, 2)	1	1	1	0	0	0
(2, 3)	0	1	1	1	1	0
(3, 4)	0	0	1	0	1	1

situation, a binary variable γ_p^k is introduced to denote the relationship between product p and line k. In particular, $\gamma_p^k = 1$, if the transport service p can be delivered by the line k, that is $(i, j) \in (IJ)^k$, and zero otherwise.

Time is discrete, there are T booking periods indexed by t, which runs forward; consequently, $t = 1$ is the first possible booking time.

At each time period $t = 1, \ldots, T$ of the booking horizon, the company has to decide on accepting/denying the request of a customer asking for a product p, that is an OD transport service.

The main goal is to maximize the total revenue coming from the accepted requests on the booking horizon.

The capacity of the system depends on the number of available seats for each line $k = 1, \ldots K$ and can be described by the following matrix:

$$X(t) = \begin{pmatrix} x_1^1 & \cdots & x_1^k & \cdots & x_{1)}^K \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{S^1}^1 & \cdots & x_{S^k}^k & \cdots & x_{S^K}^K \end{pmatrix}$$

whose generic column $x^k = (x_1^k, \ldots, x_{S^k}^k)^T$ represents the resource availability (i.e., the number of available seats) for each leg $s = 1, \ldots, S^k$ of the line k.

It is assumed that, in each time-period t of the booking horizon, at most one request for a transport service can arrive. Let λ_p^t denote the probability that, at time t, one booking request for a product p is made.

It holds that $\lambda_0^t + \sum_{t=1}^{T} \sum_{p=1}^{|(IJ)|} \lambda_{ij}^t = 1$, where λ_0^t represents the probability that no booking request arrives at time t.

Let us introduce the boolean variables μ_p^{tk}, with $\mu_p^{tk} = 1$ if and only if a request for a product p is satisfied at time t with line k. Let R_p be the revenue obtained by satisfying a request for a product p. The problem can be formulated as a dynamic program by letting $V_t(X)$ be the maximum expected revenue obtainable from periods $t, t + 1, \ldots, T$ given that, at time t, the system capacity is X.

The Bellman equation for $V_t(X)$ is reported in what follows:

$$V_t(X) = \sum_{p=1}^{|(IJ)|} \lambda_p^t \max_{\substack{\mu_p^{tk} \in \{0,1\} \\ \{k:\, \gamma_p^k = 1\}}} \left[R_p \mu_p^{tk} + V_{t+1}(\tilde{X}) \right] + \lambda_0^t V_{t+1}(X)$$

with boundary conditions:

$$V_t(0) = 0, \ \forall t;$$

$$V_{T+1}(X) = 0, \quad \text{if } x_s^k \geq 0 \text{ for all } s = 1, \ldots, S^k, \text{ for all } k$$

$$V_t(X) = -\infty, \quad \text{if } x_s^k < 0 \text{ for some } s \in S^k, k$$

We denoted by $\tilde{X}$ the matrix obtained by appropriately updating the system capacity. It is worth noting that the update of the capacity is related to following event: at time t a request for a product p occurs. The passenger transport company can accept or deny the current request. If the request is accepted by using product p^k of the line k, we need to update the leg capacities by decreasing the seats availability on all the legs involved in p^k.

4 A Linear Programming Approximation: The Bus Seat Assignment Problem

The proposed dynamic programming model is unlikely to be solved optimally due to the curse of dimensionality. In what follows, we present a linear programming approximation of the problem, that is the bus seat assignment problem (BSAP, for short) that can be used to define several revenue management policies. Starting from the dynamic programming problem, in the linear programming approximation, we replace stochastic quantities by their mean values and we assume that capacity and demand are continuous. Let be:

- d the random cumulative future demand at time t, and $\bar{d}$ its mean. In particular, d_p is the aggregate number of requests for the product p;
- R_p the revenue associated with the product p;
- y_p^k the number of products produced by the line k used to satisfy the demand for the product p;
- γ_p^k, $p = 1, \ldots, |(IJ)|$, if the product p can be delivered by using the line k and zero otherwise;
- $b_{sp^k}^k$, $s = 1, \ldots, S^k$, $p^k = 1 \ldots, |(IJ)^k|$ equal to 1 if the leg s is used in the product p^k and zero otherwise. $b_{sp^k}^k$ is an element of the matrix B introduced in the previous section.
- the resource matrix X, whose generic column vector k, $x^k = (x_1^k, \ldots, x_{S^k}^k)^T$, associated to the line k, $k = 1, \ldots, K$ indicates the number of available seats for each leg $s = 1, \ldots, S^k$ of the line k.

The total revenue achievable by the passenger transport company at time t, when the system capacity is X can be determined by solving the following optimization problem:

$$Max \sum_{p=1}^{|(IJ)|} R_p \sum_{k \in K} \gamma_p^k y_p^k \tag{1}$$

$$\sum_{k=1}^{K} \gamma_p^k y_p^k \leq \bar{d}_p \qquad p = 1, \ldots, |(IJ)| \tag{2}$$

$$\sum_{p^k=1}^{|(IJ)^k|} b^k_{sp^k} \sum_{p=1}^{|(IJ)|} \gamma^k_p y^k_p \leq x^k_s \quad k = 1, \ldots, K, \ s = 1, \ldots, S^k \tag{3}$$

$$y^k_p \geq 0, \text{integer} \qquad p = 1, \ldots, |(IJ)|, k = 1, \ldots, K \tag{4}$$

The objective function (1) represents the total revenue obtainable at time t when the residual capacity of the system is X. Constraints (2) state that the demand for a transport service p can be satisfied with all the products generated by all the lines k, $k = 1, \ldots, K$, that can deliver the considered transport service. Equations (3) control the seat availability for each line $k = 1, \ldots, K$, finally, constraints (4) define the variables domain.

5 Computational Experiments

In this section, we present the computational results obtained by solving the model described in Sect. 4, using AIMMS 4.75.3.6 and the commercial solver Cplex 10.1, on an Intel Core i7-8565U CPU, 1.8 GHz, 8 GB of RAM.

Instances For our computational tests, we have considered a real Italian case study. In particular, the instances used in this study are based on real data derived from www.simetbus.it, the web site of the Simet S.p.A., a bus company operating in the south of Italy. Each instance is characterized by a given number of lines, with a capacity of 50 seats, and each line is composed of a given set of origin-destination products or rides. A price is associated to each product. Lines, products and prices are taken from the Simet on-line website. Table 1 shows the features of the instances.

We considered three lines and 55 cities (origin or destination) numbered in ascending order. Lines are depicted on the columns. Each line is characterized by a set of OD products. In particular, the first line has 17 OD products, reported on the rows, with different origins but the same destination (i.e., 53). Line 2 is composed of 69 OD products, while Line 3 of 57. Looking at Table 1 we may observe that the lines have in common some rides. For example, the product 4–53 is present in all the lines considered; hence, departing from the origin 4, it is possible to reach the destination 53 by using all the three lines. The demand value for each ride is estimated on the basis of a load factor, defined as the ratio between demand and capacity of the line. In particular, assuming three values of load factor (1; 1.5; 1.8) the demands are estimated multiplying the load factor and the capacity.

Numerical Results In our computational study, we compare two different seat allocation strategies. In the former, we do not use the optimization model described in Sect. 4, and the demand is allocated following a first in first out (FIFO) strategy; hence, the requests are allocated following the arrival order. In the latter, the demand is allocated on the basis of solution of the BSAP model. The collected results are

Table 1 Characteristics of instances: lines and OD products

Line1		Line2						Line3					
Origin	Destination	Origin	Destination	Origin	Destination	Origin	Destination	Origin	Destination	Origin	Destination	Origin	Destination
4	53	1	51	1	52	1	53	1	53	1	54	1	55
5	53	2	51	2	52	2	53	2	53	2	54	2	55
6	53	3	51	3	52	3	53	3	53	3	54	3	55
7	53	4	51	4	52	4	53	4	53	4	54	4	55
11	53	5	51	5	52	5	53	5	53	5	54	5	55
12	53	6	51	6	52	6	53	6	53	6	54	6	55
13	53	7	51	7	52	7	53	11	53	11	54	11	55
14	53	8	51	8	52	8	53	12	53	12	54	12	55
15	53	11	51	11	52	11	53	13	53	13	54	13	55
16	53	12	51	12	52	12	53	14	53	14	54	14	55
17	53	13	51	13	52	13	53	15	53	15	54	15	55
18	53	14	51	14	52	14	53	16	53	16	54	16	55
20	53	15	51	15	52	15	53	17	53	17	54	17	55
21	53	16	51	16	52	16	53	18	53	18	54	18	55
22	53	17	51	17	52	17	53	20	53	20	54	20	55
31	53	18	51	18	52	18	53	22	53	22	54	22	55
32	53	19	51	19	52	19	53	31	53	31	54	31	55
		20	51	20	52	20	53	32	53	32	54	32	55
		21	51	21	52	21	53			53	54	53	55
		22	51	22	52	22	53					54	55
		31	51	31	52	31	53						
		32	51	32	52	32	53						
				51	52	51	53						
						52	53						

Table 2 Computational results—Total revenue: BSAP vs FIFO

Load factor	BSAP [€]	FIFO [€]
1	4659	4402
1.5	5258	4672
1.8	5361	4979

provided in Table 2, where the first column reports the load factor, the second one the total revenues obtained by solving the BSAP model, and the third column shows the revenue obtained using the FIFO strategy. All the revenues are expressed in €.

Looking at results depicted in Table 2, it is easy to notice that overall BSAP is the most effective strategy. In fact, we may observe an increase in the revenue, compared to the FIFO strategy, that is about the 5.8%, 12.5% and 7.7% when using a load factor equal to 1, 1.5 and 1.8, respectively. In addition, solving the model does not require high computational time, about 0.05 s on average. The BSAP model allows to efficiently and effectively manage the seats assignment, allocating the products with higher price, in order to maximize the total revenue. Indeed, the majority of allocated seats are the most expensive ones. Hence, the strategy adopted is selling less tickets but for the seats with the highest prices. To conclude, it is evident that the application of the BSAP model, together with a revenue management policy based on the dynamic determination of the price, could help the bus transport company to achieve more suitable solutions in terms of revenue.

6 Conclusions and Future Work

In this paper, we considered the bus seat allocation problem to optimize the allocation of customers demand and maximize the overall revenue. In the computational experiments, we made a comparison between the results obtained with and without the application of the proposed model. The results highlighted that our model provides more effective solutions than a common FIFO strategy. This model could help the bus transport company to optimize the managing of resource-constrained buses and, at the same time, maximizing the revenue. It is worth noting that the bus seat assignment problem can be used as the baseline to define several revenue management policies that can be applied and compared in future works, such as: booking limits policy, bid price policy and opportunity cost policy. Furthermore, the proposed model could represent a base for several interesting extensions. It is worth noting that in our framework we considered a single line performed once per day, thus, different departure times for the same destination are not taken into account. Since the companies usually give the possibility to choose the departure time, an interesting extension of this model could take into account this feature.

References

1. Ammirato, S., Felicetti, A.M., Linzalone, R., Volpentesta, A.P., Schiuma, G.: A systematic literature review of revenue management in passenger transportation. Meas. Bus. Excell. **24**(2), 223–242 (2020)
2. Ciancimino, A., Inzerillo, G., Lucidi, S., Palagi, L.: A mathematical programming approach for the solution of the railway yield management problem. Transp. Sci. **33**, 168–181 (1999)
3. Dror, M., Trudeau, P., Ladany S.P.: Network models for seat allocation on flights. Transp. Res. B Methodol. **22**(4), 239–250, 1988.
4. Gallego, G., Phillips, R.: Revenue management of flexible products. Manuf. Serv. Oper. Manag. **6**, 321–337 (2004)
5. Guerriero, F., Olivito, F.: Revenue models and policies for the car rental industry. J. Math. Model. Algorithms Oper. Res. **13**, 247–282 (2014)
6. Gönsch, J., Steinhardt, C.: Using dynamic programming decomposition for revenue management with opaque products. Bus. Res. **6**(1), 94–115 (2013)
7. Hetrakul, P., Cirillo, C.: A latent class choice based model system for railway optimal pricing and seat allocation. Transp. Res. E Logist. Transp. Rev. **61**, 68–83 (2014)
8. Hu, X., Shi, F., Xu, G., Qin, J.: Joint optimization of pricing and seat allocation with multistage and discriminatory strategies in high-speed rail networks. Comput. Ind. Eng. **148**, 106690 (2020)
9. Klein, R., Koch, S., Steinhardt, C., Strauss, A.K.: A review of revenue management: Recent generalizations and advances in industry applications. Eur. J. Oper. Res. **284**(2), 397–412 (2020)
10. Ongprasert, S.: Passenger behavior on revenue management systems of inter-city transportation. Technical report, Graduate School of Engineering. Kochi University of Technology, Japan, 2006
11. Sierag, D.D., Koole, G.M., van der Mei, R.D., van der Rest, J.I., Zwart, B.: Revenue management under customer choice behaviour with cancellations and overbooking. Eur. J. Oper. Res. **246**(1), 170–185 (2015)
12. Subramanian, J., Stidham, S., Lautenbacher, C.J.: Airline yield management with overbooking, cancellations, and no-shows. Transp. Sci. **33**(2), 147–167 (1999)
13. Tang, C., Ceder, A., Ge, Y.E., Wu, N.: Optimal operational strategies for multiple bus lines considering passengers' preferences. Transp. Res. Rec. **2674**(5), 572–586 (2020)
14. Vinod, B.: Evolution of yield management in travel. Revenue Pricing Manag. **15**, 203–211 (2016)
15. Wang, Y., Lan, B.X., Zhang, L.: A revenue management model for high-speed railway. In: Ye, X.W., Ni, Y.Q. (eds.) Proceedings of the 1st International Workshop on High-Speed and Intercity Railways, vol. 147, pp. 95–103. Springer (2012)
16. Wang, X., Wang, H., Zhang, X.: Stochastic seat allocation models for passenger rail transportation under customer choice. Transp. Res. E Logist. Transp. Rev. **96**, 95–112 (2016)
17. Wittman, M.D., Belobaba, P.P.: Dynamic availability of fare products with knowledge of customer characteristics. J. Revenue Pricing Manag. **16**(2), 201–217 (2017)